PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/32	A2	(11) International Publication Number: WO 96/12362 (43) International Publication Date: 25 April 1996 (25.04.96)

(21) International Application Number: PCT/NL95/00350

(22) International Filing Date: 12 October 1995 (12.10.95)

(30) Priority Data:
08/321,855 14 October 1994 (14.10.94) US

(71)(72) Applicant and Inventor: BRANDS, Stefanus, Alfonsus [NL/NL]; Ina Boudier-Bakkerlaan 143 III, NL-3582 XW Utrecht (NL).

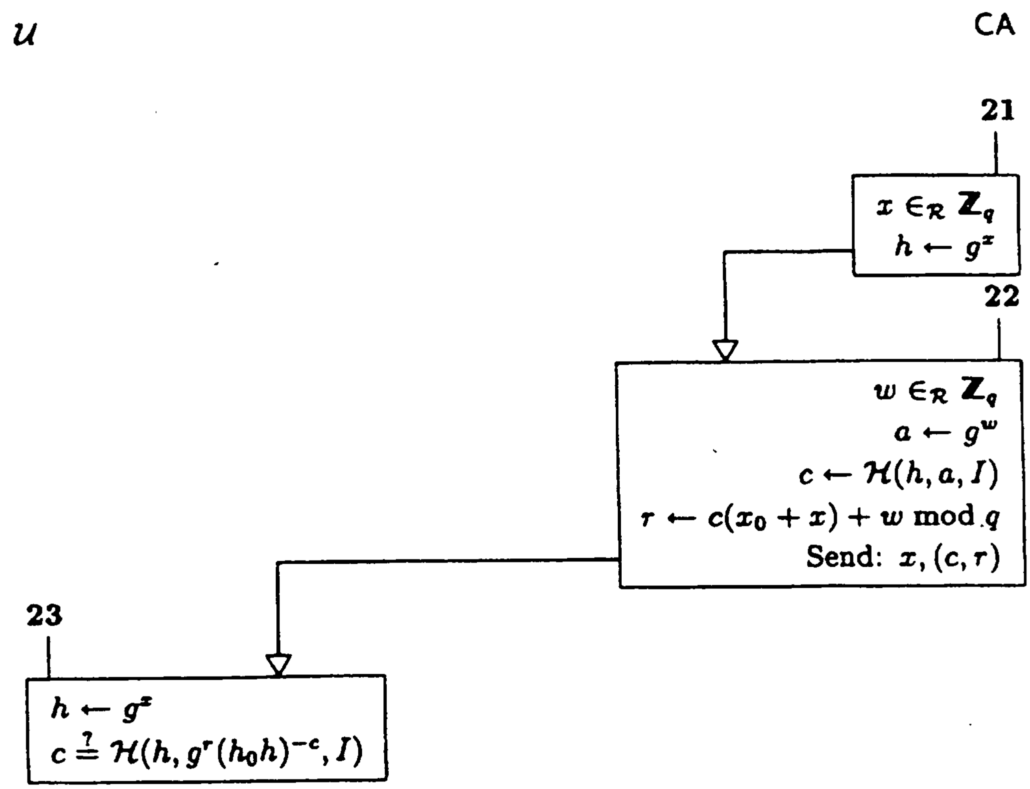
(81) Designated States: AM, AU, BB, BG, BR, BY, CA, CN, CZ, EE, FI, GE, HU, IS, JP, KG, KP, KR, KZ, LK, LR, LT, LV, MD, MG, MN, MX, NO, NZ, PL, RO, RU, SG, SI, SK, TJ, TM, TT, UA, UG, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ, UG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: SECRET-KEY CERTIFICATES

(57) Abstract

Cryptographic methods and apparatus are disclosed that enable formation and issuance of secret-key certificates. In contrast the well-known cryptographic technique of public-key certificates, where a public-key certificate is a digital signature of a certification authority on a public key, pairs consisting of a public key and a corresponding secret-key certificate can be generated by anyone. However, triples consisting of a secret key, a matching public key and a secret-key certificate corresponding to the public key, can only be generated when the certification authority is involved. Also described are applications of secret-key certificates to public-key directories and to restrictive blind issuing protocols.



BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgystan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SI	Slovenia
CI	Côte d'Ivoire	KZ	Kazakhstan	SK	Slovakia
CM	Cameroon	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
CZ	Czech Republic	LV	Latvia	TJ	Tajikistan
DE	Germany	MC	Monaco	TT	Trinidad and Tobago
DK	Denmark	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali	UZ	Uzbekistan
FR	France	MN	Mongolia	VN	Viet Nam
GA	Gabon				

SECRET-KEY CERTIFICATES

BACKGROUND OF THE INVENTION

1. Field of the invention.

The present invention relates to cryptographic techniques, and more particularly to methods and apparatus for implementing certificate schemes based on public-key cryptography.

2. Description of the prior art.

Public-key certificates, usually plainly referred to as certificates, are an important cryptographic tool for secure key management. The idea is to have a specially appointed party, commonly called the Certification Authority, certify the public keys of other parties in the system by digitally signing these public keys with its own secret key. By widely distributing the public key of the Certification Authority through a variety of media, anyone can be assured that it is genuine. Because a public-key certificate is a digital signature of the Certification Authority on a public key, certificates on public keys of other parties can be verified by anyone by using the public key of the Certification Authority. The net effect is that impersonation attacks, and similar other attacks, are prevented.

In practical applications, the certificates of the Certification Authority may, and perhaps should, certify additional information. Along with the public key, a certificate could validate such information as the name of the party associated with the public key, employer, telephone number, electronic mail address, and a list of access rights.

To facilitate practicality of the certificate issuing process, public keys can be certified recursively, according

to a hierarchical structure. For example, in an electronic cash system, the main bank can certify the public keys of all the local banks, the local banks in turn can certify public keys in POS terminals by using their certified keys, and the secret keys corresponding to the public keys in the POS terminals can be used to decrypt information that is sent by the host. The hierarchic certification process can be thought of as building a tree, each node containing a public key and a certificate on the public key. A certificate on a public key in a node is a digital signature on that public key, that has been computed by the party associated with the parent node by applying the secret key that corresponds to the public key of the parent node. Anyone can verify the validity of a public key by recursively descending (or ascending) the tree from the root node to the node associated with the public key that is being verified (or vice versa). A certification hierarchy that is often suggested is one that is implied by the lifetime of the cryptographic keys: keys that are more susceptible to attacks are changed more frequently, and are certified by keys that have a longer lifetime.

Public keys can be listed in so-called public-key directories, which can be made available on CD-ROM or other media. In order to encrypt a message intended for another party, one merely needs to look up the public key of that other party in the public-key directory, verify the validity of the certificate, and encrypt the message with the public key. It can then be sent to the other party. No interaction is needed between the two parties. In this way for instance encrypted electronic mail can be sent over a computer network.

Because the certificate mechanism obviates the need for the public-key directory to be secured, public keys need not necessarily be listed in a public-key directory. They may be sent (along with the certificate) on request, by the party associated with the public key itself, or by any other party that need not be trusted, such as a server in a computer network.

In cryptographic mechanisms for transfer of credentials, the Certification Authority at issuing time can issue a

certificate on a public key of a user. The type of credential that is issued can, for instance, be denoted by the type of signature that the Certification Authority computes. This allows the user, when transferring the credential to a recipient, to make a digital signature on a message of the recipient (describing such information as the identity of the recipient and transaction details), by using the secret key corresponding to his certified public key. The certificate proves the validity of the credential to the recipient, whereas the signature made by the user proves that the user willingly transferred the credential to the recipient.

For privacy-protected transfer of credentials, the information that is issued by the Certification Authority should not be linkable to executions of the issuing protocol. Special techniques are known that enable the user to blind the issuing protocol while interacting with the Certificate Authority.

While important and useful, the public-key certificate technique also has a few problems associated with it. First of these relates to privacy. It is conceivable that providers for a variety of electronic systems available in the near future will require participants to meet certain criteria before certifying their public keys. These criteria may include social status, income, type of job, trustworthiness, and so on. Because a public-key certificate is a digital signature of the Certification Authority on the public key, pairs consisting of a public key and a corresponding public-key certificate reveal to anyone which parties are participating in a certain system, and which parties are not participating. This reveals which parties meet the criteria specified by the Certification Authority, and which parties may not meet them. Likewise, the genuineness of the additional information (employer, telephone number, access rights, and so on) that may have been certified along with the public key, is revealed. Consequently, public-key certificates allow anyone to extract profiles of other parties, by scanning for their appearances, or the lack thereof, in compiled lists of certified public keys (such as

public-key directories). This problem is by no means removed by letting participants send their public keys only on request, instead of using a public-key directory.

A second problem is that the publication of a public-key directory reveals a huge amount of digital signatures of the Certification Authority on known, or chosen, public keys. Although most of the known digital signature schemes are believed to be secure under known, or (adaptively) chosen, message attacks, only a few signature schemes are known that can be proven to be secure, assuming the existence of functions that are substantially unfeasible to invert. Unfortunately, these schemes are currently not practical for large-scale use. Since public-key directories typically will contain an enormous amount of entries, the Certification Authority will have to use an efficient signature scheme. This implies that the signatures in the public-key directory may be helpful in attempts to break the signature scheme of the Certification Authority; they can be used to mount known or (adaptively) chosen message attacks. Again, this problem is not removed by letting participants send their public keys only on request, instead of using a public-key directory.

A third problem is in blinding public-key certificate issuing protocols in mechanisms for privacy-protected transfer of credentials (see, for instance, U.S. Patent No. 4,759,063 to Chaum for a discussion of the technique of blinding in public-key cryptography). In many circumstances, the Certification Authority does not want the users to be able to blind to their hearts' contents, but would like to encode information in the issued information that cannot be changed by the blinding operations of the user. For instance, in mechanisms for transferring credentials under pseudonym, this encoded information can be uniquely associated with the user that the credential is issued to, thereby linking the pre-images of all the pseudonyms of each user. In this way, it can be ensured that users cannot use the credentials of other users, even if they cooperate. For credentials that may be shown only a limited number of times, such as coins in an electronic cash system, it can be arranged that this encoded

information is revealed if and only if the credential is shown a number of times exceeding a predetermined limit. This obviates the need for on-line verification of these credentials. For such purposes, an issuing protocol is needed in which the Certification Authority issues a secret key, a public key, and a public-key certificate, in such a way that the public key and the certificate can be blinded by the user, but a non-constant function of the secret key cannot. Such an issuing protocol is called a restrictive blind signature issuing protocol, and is described and claimed in patent application Ser. No. PCT/NL94/00179, filed August 1, 1994, and is incorporated by reference herein. From the point of view of security, no satisfactory constructions of restrictive blind signature issuing protocols are known in which the certificate is a public-key certificate. This is a serious problem, since restrictive blind signature issuing protocols are of crucial importance for the construction of efficient and secure mechanisms for privacy-protected off-line transfer of credentials.

Patent application Ser. No. PCT/NL94/00179, filed August 1, 1994, also describes and claims an inventive method for constructing restrictive blind signature issuing protocols where the issued certificate is not a digital signature on the public key (and hence not a public-key certificate). As is demonstrated in detail, the construction of efficient and secure restrictive blind signature issuing protocols becomes much easier by removing the need for the certificate to be a signature of the issuer on the public key. Most (more specifically, all but the last one described) of the exemplary restrictive blind signature issuing protocols described and claimed in patent application Ser. No. PCT/NL94/00179 are constructed by applying this inventive method.

While the inventive method described and claimed in patent application Ser. No. PCT/NL94/00179 overcomes the third problem associated with public-key certificates, it does not address the first two problems. This invention describes a generalized method that overcomes all three problems associated with public-key certificates.

OBJECTS OF THE INVENTION

Accordingly, it is an object of the present invention to allow anyone to generate pairs consisting of a public key and a corresponding certificate, while at the same time ensuring that it is unfeasible to generate, without knowledge of a special secret key that is held by a Certification Authority, triples consisting of a secret key, a matching public key, and a corresponding certificate, by providing for a new kind of certificates that will henceforth be called secret-key certificates.

Another object of the present invention is to prevent lists of certified public keys, such as public-key directories, from revealing the genuineness of privacy-related information, by using secret-key certificates instead of public-key certificates.

A further object of the present invention is to prevent lists of certified public keys, such as public-key directories, from revealing information that may be helpful in known or chosen message attacks on the signature scheme of the Certification Authority, by using secret-key certificates instead of public-key certificates.

Yet another object of the present invention is to describe techniques to construct secret-key certificates, and issuing protocols therefor, from well-known digital signature schemes that are commonly referred to in the art as Fiat/Shamir type signature schemes (see, Fiat, A. and Shamir, A., "How to prove yourself: practical solutions to identification and signature problems," Crypto '86, Springer-Verlag (1987), pp. 186-194).

A still further object of the present invention is to construct efficient and secure restrictive blind secret-key certificate issuing protocols, by letting the Certification Authority issue triples consisting of a secret key, a matching public key, and a corresponding secret-key certificate, such that the public key and the certificate can be blinded by the receiving party, but at least part of the secret key cannot be blinded.

An even further object of the present invention is to implement hierarchical certification by recursive application of secret-key certificates instead of public-key certificates.

Still another object of the present invention is to allow efficient, economical, and practical apparatus and methods fulfilling the other objects of the invention.

Other features, objects, and advantages of this invention will be appreciated when the description and appended claims are read in conjunction with the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure **1** shows a representative combination block and functional diagram of an exemplary secret-key certificate system in accordance with the teachings of the present invention.

Figure **2** shows a flowchart of a secret-key certificate issuing protocol for a first preferred embodiment in accordance with the teachings of the present invention.

Figure **3** shows a flowchart of a secret-key certificate issuing protocol, such that the Certification Authority does not need to know the secret key of the recipient, for the first preferred embodiment in accordance with the teachings of the present invention.

Figure **4** shows a flowchart of a secret-key certificate issuing protocol, such that the subliminal channel in the secret-key certificate is prevented, for the first preferred embodiment in accordance with the teachings of the present invention.

Figure **5** shows a flowchart of a secret-key certificate issuing protocol, such that the recipient fully blinds the issued information, for the first preferred embodiment in accordance with the teachings of the present invention.

Figure **6** shows a first flowchart of a restrictive blind secret-key certificate issuing protocol for the first preferred embodiment in accordance with the teachings of the present invention.

Figure **7** shows a second flowchart of a restrictive blind secret-key certificate issuing protocol for the first

preferred embodiment in accordance with the teachings of the present invention.

Figure **8** shows a flowchart of a secret-key certificate issuing protocol for a second preferred embodiment in accordance with the teachings of the present invention.

Figure **9** shows a flowchart of a secret-key certificate issuing protocol, such that the Certification Authority does not need to know the secret key of the recipient, for the second preferred embodiment in accordance with the teachings of the present invention.

Figure **10** shows a flowchart of a secret-key certificate issuing protocol, such that the subliminal channel in the secret-key certificate is prevented, for the second preferred embodiment in accordance with the teachings of the present invention.

Figure **11** shows a flowchart of a secret-key certificate issuing protocol, such that the recipient fully blinds the issued information, for the second preferred embodiment in accordance with the teachings of the present invention.

Figure **12** shows a flowchart of a restrictive blind secret-key certificate issuing protocol for the second preferred embodiment in accordance with the teachings of the present invention.

SUMMARY OF THE INVENTION

In accordance with these and other objects of this invention, a brief summary of the invention is presented. Some simplifications and omissions may be made in the following summary, which is intended to highlight and introduce some aspects of the present invention, but not to limit its scope. Detailed descriptions of preferred exemplary embodiments adequate to allow those of ordinary skill in the art to make and use this invention will be provided later.

In essence, the primary purpose of a public-key certificate is to certify the tasks that a participant in a cryptographic system can successfully perform with respect to his public key, rather than the public key itself. Consider the three most well-known public-key cryptographic tasks in a

cryptographic system: digital signing, identification, and encryption. The intended purpose of a certificate is to certify that the tasks of digital signing, proving knowledge of a secret key that corresponds to a public key, and decrypting a message that is encrypted with a public key, are performed using a secret key the usage of which has been permitted by the Certification Authority. In other words, it is the secret key that must be certified, not necessarily the public key.

Informally, a secret-key certificate is a digital signature of the Certification Authority on a secret key, such that it is not a digital signature on the public key that matches the secret key. More precisely, triples consisting of a secret key, a matching public key, and a corresponding secret-key certificate can be feasibly generated only by the Certification Authority, but pairs consisting of a public key and a corresponding secret-key certificate can be feasibly generated by anyone.

In a public-key directory, parties are listed together with their public keys and corresponding certificates. If the certificates are secret-key certificates, then the information in the directory does not reveal discriminating information about legitimate participation, as each of the entries could have been generated by anyone (if the directory is, or has been at a certain point in time, open for public writing). If additional information is included as well, such as telephone numbers, postal address, and access rights, then this may be bogus information as well. To ensure that users will be unable to tell whether entries in a public-key directory have been certified by the Certification Authority or not, the manufacturer of a public-key directory on, say, a CD-ROM, can gather the entries in the Directory by letting the parties associated with the public keys submit their own public keys and associated secret-key certificates, as they wish them to appear on the CD-ROM.

For the same reason, the information listed in the directory cannot be of much help in attacking the signature scheme of the Certification Authority. In particular, it can be of no

help whatsoever if pairs consisting of a public key and a corresponding secret-key certificate can be generated, without cooperation of the Certification Authority, with a probability distribution that is indistinguishable from the probability distribution that applies when the certificate issuing protocol is performed with the Certification Authority.

An attacker whose keys have not been certified by the KAC would need to know the secret keys of legitimate participants in the system in order to have any advantage in breaking the signature scheme of the Certification Authority over trying to break it from scratch. Revealing one's secret key brings along a great deal of trust in that it will not be misused, and so in practice the consequence of using secret-key certificates instead of public-key certificates is that attacks to break the signature scheme of the KAC will be much harder to mount.

Secure public-key cryptographic schemes, such as schemes for the aforementioned tasks of proving knowledge of a secret key corresponding to a public key, signing a message with a secret key corresponding to a public key, and decrypting a message encrypted with a public key, are feasible to perform only if one actually knows the secret key corresponding to the public key. Hence, the fact that a user can successfully perform such a task attests to the fact that he knows the secret key corresponding to his public key, and this in turn proves that the certificate must have been issued by the Certification Authority.

The following illustrations are helpful to appreciate the use of secret-key certificates:

1. Suppose that a first party wants to transfer an encrypted message to a second party. From the public-key directory, the first party retrieves the public key of the second party, and a corresponding secret-key certificate, which supposedly has been issued by the Certification Authority. The first party encrypts its message using the public key of the second party, and transfers it to the second party. Although the first party does not know whether the information listed in the public-key directory is genuine or

not, it is ensured that if the second party can decrypt and retrieve the original message, then the Certification Authority computed the string listed in the public-key directory. Obviously, a proper response of the second party to the first party will allow the first party to distinguish between the two cases.

2. Suppose that the second party digitally signs a message for the first party. Given the message, the digital signature of the second party, and the information listed with the second party in the public-key directory, the first party is able to verify not only that the digital signature is genuine, but it is also ensured that it was indeed made with a certified key. Of course, if this signature is to have any legal significance, then anyone should be able to verify these two facts.

3. Suppose that the second party wishes to obtain a secret-key certificate, issued by the first party. To this end, the first party issues, by applying its secret key, a secret-key certificate on a public key of the second party. Not only can the second party verify the validity of the secret-key certificate issued by the first party, by using the public key of the first party, but, if the verification holds, it also is ensured that the secret-key certificate on the public key of the first party has been issued by the Certification Authority. In general, recursive application of secret-key certificates allows hierarchical certification trees to be formed that offer the same security as hierarchical certification trees formed from public-key certificates.

An example of a secret-key certificate and a corresponding issuing protocol will now be described. As is common in the art, for any integer, l, the symbol $\mathbb{Z}_l$ denotes the set of integers $\{0,\ldots,l-1\}$, for which addition and multiplication are defined modulo l. Let G_q denote a group of prime order, q, for which no feasible algorithms are known to compute discrete logarithms, and let $\mathcal{H}$ be a one-way hash-function that maps its arguments to $\mathbb{Z}_{2^t}$, where t is an appropriately large security parameter. The secret key of the Certification Authority is a

number x_0 in $\mathbb{Z}_q$, and the corresponding public key is (g, h_0), where h_0 denotes g^{x_0}, and both g and h_0 are in G_q. The parties whose keys are to be certified use the same key set-up as the Certification Authority: the secret key of a party $\mathcal{U}$ is a number x in $\mathbb{Z}_q$, and the corresponding public key is $h = g^x$. A secret-key certificate on h is a pair (c, r), where c is in $\mathbb{Z}_{2^t}$ and r is in G_q, such that c is equal to $\mathcal{H}(h, g^r(h_0h)^{-c}, I)$. Here, I is an, optionally included, string consisting of additional information about the party associated with the public key.

As will be clear to those of ordinary skill in the art, the certificate of the Certification Authority in effect is a Schnorr signature (see; Schnorr, C., "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol. 4, No. 3, (1991), pp. 161-174) on g^x, made with secret key $x_0 + x \bmod q$. Anyone can feasibly generate pairs $h, (c, r)$ such that the verification relation holds, by taking h equal to $h_0^{-1}g^s$ for an arbitrary s in $\mathbb{Z}_q$, and a equal to g^t for an arbitrary t in $\mathbb{Z}_q$; the pair $(c, sc + t \bmod q)$, with $c = \mathcal{H}(h, a, I)$, then matches h. However, the ability to feasibly generate such pairs together with the secret key $\log_g h$ would enable one to forge Schnorr signatures.

To issue a secret-key certificate, the Certification Authority can proceed as follows. It generates at random a number w in $\mathbb{Z}_q$, and computes $a = g^w$. It then computes $c = \mathcal{H}(h, a, I)$ and $r = c(x_0 + x) + w \bmod q$, and transfers (c, r) to $\mathcal{U}$. As will be demonstrated in the detailed description, the issuing protocol can be modified such that the need for the Certification Authority to know the secret key of $\mathcal{U}$ can be prevented. Moreover, a variety of blinding capabilities for $\mathcal{U}$ can be incorporated into the issuing protocol.

In sum, the present invention describes certificate techniques based on public-key cryptography, that solve problems related to the well-known cryptographic technique of public-key certificates.

DETAILED DESCRIPTION OF THE INVENTION

While it is believed that the notation of FIGS. **2** to **12** would be clear to those of ordinary skill in the art, it is

first reviewed here for definiteness.

A variety of secret-key certificate issuing protocols is described by flowcharts. The actions performed by the parties participating in these protocols are grouped together into flowchart boxes. The party performing the actions described in a flowchart box is indicated by the column that the box is in. The columns are labeled by a symbol denoting the type of party. The term "party" is used to indicate an entity that might sometimes be regarded as an agent who performs a step or a collection of steps in a protocol. It might also be regarded as a means for performing those steps, and might be comprised of any suitable configuration of digital logic circuitry. For example, any box or collection of boxes from the figures could be realized by hard-wired and dedicated combinatorial logic, or by some sort of suitably programmed machine, a microprocessor for instance, such as are well-known in the art, just as long as it is able to perform the storage, input/output and transformational steps (possibly apart from the random source functions) described by the corresponding box or boxes.

As is common in the art, for any integer, l, the symbol $\mathbb{Z}_l$ denotes the set of numbers $\{0,\ldots,l-1\}$. Addition and multiplication of elements in $\mathbb{Z}_l$ are defined modulo l. The symbol $\mathbb{Z}_l^*$ denotes the set of numbers in $\{0,\ldots,l-1\}$ that are co-prime to l. Multiplication of elements in $\mathbb{Z}_l^*$ is defined modulo l. As is well-known in the art, $\mathbb{Z}_l$ is called a ring of integers modulo l, and $\mathbb{Z}_l^*$ is called a multiplicative group of integers modulo l.

The symbol "$\leftarrow$" denotes assignment, meaning that the variable or symbol on its left-hand side is assigned the value on its right-hand side to. The assignments do not necessarily imply that storage space must actually be reserved; they may indicate intermediate values manipulated in volatile memory.

Another operation is a test for equality, indicated by the $\stackrel{?}{=}$ symbol. As is common in the art, the protocol halts in case the equality does not hold.

The symbol $\in_{\mathcal{R}}$ indicates that the number, or each of the numbers, on its left-hand side is chosen from the set on its

right-hand side according to a uniform probability distribution, and independent of anything else. In practice, pseudo-random techniques may be used, and the deviation from the uniform distribution may be significant without resulting in an appreciable loss in security and/or privacy. Such techniques are well-known in the art.

Another action is denoted by the word "Send," followed by a colon and one or more numbers. This indicates that these numbers are sent by the party performing the actions described in the box to the other party participating in the protocol. The directed connections between the boxes indicate the order in which the actions that are grouped in the boxes are performed.

Apparatus for Secret-Key Certificates Systems.

A precise description of the apparatus for a secure secret-key certificate system will now be given. The apparatus comprises the following five means:

1. First key generation means that, on being given as input at least a security parameter, outputs a secret key and a matching public key, to be used by the Certification Authority for certifying keys of parties that wish to participate in the cryptographic system.

As will be clear to those of o rdinary skill in the art, the binary length of the output is polynomially (preferably linearly) related to the security parameter.

The first key generation means is of a probabilistic nature, meaning that the key pair is generated in a substantially random manner. Preferably, the randomization process is based on the output of some physical source of randomness, possibly combined with the output of a cryptographically strong pseudo-random number generator by taking, for instance, the bitwise exclusive-or of the two outputs.

2. Second key generation means that, on being given as input at least a security parameter, outputs a key pair consisting of a secret key and a matching public key, to be used by a party that wishes to participate in the cryptographic system.

As with the first key generation means, the output is the result of a suitable randomization process. The means may be operated by the party itself, by the Certification Authority, or by the both of them. The first and second key generation means may, but need not, be identical.

3. Certificate verification means that, on being given as input the public key of the Certification Authority and a pair consisting of a public key and a presumed secret-key certificate, outputs a response that is to be interpreted as "yes" or "no."

Usually, the verification relation will be deterministic, but this need not be the case; the certificate verification means may be of probabilistic nature, performing a great many random trials in order to decide whether the presumed certificate is a secret-key certificate on the public key.

The certificate verification means outputs "yes" if and only if the presumed certificate is a secret-key certificate on the public key. In other words, the certificate verification means defines what a secret-key certificate on a public key is.

4. Certificate issuing means that, on being given as input the secret key of the Certification Authority and a pair consisting of a secret key and a matching public key of a party that requires a certificate, outputs a digital signature on the secret key, such that the certificate verification means, on being given as input the public key of the Certification Authority and a pair consisting of the public key and the issued digital signature, outputs "yes" (and so the output of the certificate issuing means is a secret-key certificate on the public key).

The certificate issuing means may, but need not, be of a probabilistic nature.

Obviously, it will usually suffice to take only the secret key of the Certification Authority and the secret key of the party as input, since the public key can usually be computed from the secret key.

As will be appreciated, the certificate issuing means can be such that the party, whose keys are to be certified, can keep

secret from the Certification Authority its secret key. In that case, the certificate issuing means itself comprises means controlled by the party, means controlled by the Certification Authority, and suitable interface means for allowing communication therebetween. The means controlled by the party takes as input the secret key of the party (and possibly also the public key of the party), and the means controlled by the Certification Authority takes as input the secret key of the Certification Authority (and possibly also the public key of the party). The final output is the result of processing by the means controlled by the party and the means controlled by the Certification Authority, where appropriate intermediate results may be communicated through the interface means. The precise actions to be performed by the means that the certificate issuing means comprises, must be described by a cryptographic protocol, henceforth called a secret-key certificate issuing protocol.

Yet another variation is that the Certification Authority can compute the secret keys corresponding to any public key of a party that requires a secret-key certificate, because it knows additional trap-door information.

These and other variations will be appreciated by studying the exemplary embodiments. For example, the certificate issuing process may be blinded by the means controlled by the party, according to a variety of criteria that will be described in the text.

5. Certificate simulating means that, on being given as input the public key of the Certification Authority, outputs a pair consisting of a public key and a matching certificate.

The public key that is output is such that it could have been output by the second key generation means (as part of the pair that it outputs). "Matching" indicates that the certificate verification means, on being given as input the public key of the Certification Authority and the output of the certificate simulating means, outputs "yes." The certificate simulating means is of a probabilistic nature, and the probability distribution of its output should be substantially indistinguishable from the probability

distribution that applies when the public key is generated by the second key generation means and the certificate is generated by the certificate issuing means.

As will be clear to those of ordinary skill in the art, "substantially" indistinguishable may mean "computationally," "statistically," or "perfectly" indistinguishable, each of which has a precise mathematical meaning that is well-known in the art. Obviously, the indistinguishability property need not be this strong for practical purposes. For instance, if the set of possible outputs produced by the certificate simulating means on being given a certain input is sufficiently large, it might still be infeasible in practice to distinguish between simulated pairs and "genuine" pairs consisting of a public key and a matching secret-key certificate.

In applications in which parties wish to generate pairs consisting of a public key and a corresponding secret-key certificate, without cooperation of the Certification Authority, the certificate simulating means must obviously be constructed. However, as will be illustrated later on, the certificate simulating means need not always build.

Each of the described five means can be realized by hard-wired and dedicated combinatorial logic, or by some sort of suitably programmed machine, a microprocessor for instance, such as are well-known in the art.

A secret-key certificate on a public key, as issued by the certificate issuing means, is in effect a digital signature on the secret keys corresponding to the public key. Hence it is unfeasible for parties other than the Certification Authority to generate, without involvement of the Certification Authority, new triples consisting of a secret key, a matching public key (meaning that the pair could have been generated by the second key generation means), and a corresponding secret-key certificate (meaning that the certificate verification means would output "yes" when being given as input the public key of the Certification Authority, and the public key and the certificate).

It is stressed that the secret-key certificate is said to be

a secret-key certificate on the public key, not on the secret key: both the certificate issuing means and the certificate simulating means can output pairs consisting of a public key and a secret-key certificate on the public key. This emphasizes that there is a publicly verifiable relation between the public key and the secret-key certificate.

Turning now to FIG. **1**, an exemplary description of the interconnection and cooperation of the constituent parts described above will now be presented. All the lines in FIG. **1** imply the transfer of messages. These may be held initially or delayed on their way, encoded and decoded cryptographically or otherwise to provide their authenticity and/or secrecy and/or error detection and/or error recovery. Thus the particular means or methods whereby messages are transferred are not essential to the present invention, and it is anticipated that any technique may be employed in this regard. The lines may for example be taken to represent communication means, in which case they might be realized in a variety of exemplary ways including conductive paths, fibre optic links, infra-red transmission, or paths through a packet switched network; also suitable drivers, modems, or other appropriate interfaces may be required at the ends of such lines, as are well-known in the art. Alternatively, the lines may be taken to stand for a message transfer step.

First key generation means **113** transforms a security parameter on line **102** to a key pair for the Certification Authority. The key pair consists of a public key, output on line **106**, and a secret key, output on line **105**. The transformation of key generation means **113** depends on, amongst others, random number generator **119**.

Second key generation means **112** transforms a security parameter on line **101** to a public key, output on line **103**, and a secret key, output on line **104**. The transformation of key generation means **112** from the security parameter on line **101** to the keys on lines **103** and **104** depends on, amongst others, random number generator **118**. The outputs of the key generation means will be certified by certificate issuing means **114**, as detailed next.

Certificate issuing means **114** takes three inputs, on lines **103**, **104**, and **105**. The input on line **103** is the public key output by key generation means **112**, and the input on line **104** is the secret key output by key generation means **112**. The input on line **105** is the secret key of the Certification Authority output by key generation means **113**. Certificate issuing means **114** transforms these three inputs to a secret-key certificate on the public key of line **103**. This secret-key certificate is a digital signature on the secret key of line **104**, and is output on line **107**. Not displayed is an optional random generator in certificate issuing means **114**, although the preferred embodiments that are described in detail use randomness in the certificate issuing means.

The certificate, output on line **107**, is fed into certificate verification means **116**, together with the public key that was output by key generation means **112** on line **103**. The public key of the Certification Authority on line **106** is also fed into certificate verification means **116**. The certificate verification means outputs a binary value on line **110**, which is to be interpreted as a verdict about the correctness of the inputs on lines **103** and **107**. In the block diagram, the input on line **107** to certificate verification means **116** is the output of certificate issuing means **114**, and the input on line **103** is the same as the input to certificate issuing means **114**. Hence, the verdict on line **110** will in this case be affirmative ("yes"). Not displayed is an optional random generator in certificate verification means **116**. The preferred embodiments that are described in detail do not use randomness in the certificate verification means. It is conceivable that for certain applications it may be necessary to incorporate randomness in the certificate verification means.

The public key of the Certification Authority on line **106** is fed into certificate simulating means **115**. Depending on, amongst other, random number generator **120**, the certificate simulating means **115** transforms the input on line **106** to a public key on line **109**, and a secret key certificate on this public key on line **108**. When the two outputs of certificate

simulating means **115**, and the public key of the Certification Authority on line **106**, are fed into certificate verification means **117**, a binary value is output on line **111**. This output is a verdict about the correctness of the inputs on lines **109** and **108**, and will in the displayed situation be affirmative ("yes"). Certificate verification means **117** is the same as means **116**, as are all its input and output lines, and has been drawn twice only to emphasize that the certificate verification means cannot distinguish between a secret-key certificate output by certificate issuing means **114**, and a secret-key certificate that is output by certificate simulation means **115**.

A variety of exemplary secret-key certificate systems for each of two preferred embodiments will now be provided. The Certification Authority will henceforth be denoted by CA, and a party in the system by $\mathcal{U}$ (for "user"). In both preferred embodiments, the various secret-key certificates that will be described are constructed from a class of signature schemes that is well-known in the art, henceforth referred to as Fiat/Shamir type signatures, by applying an inventive general construction technique. (Fiat/Shamir type signature schemes are signature schemes that are derived from secure three-transmission identification schemes of the challenge-responses type, by taking the challenge as a one-way hash of at least the message and information provided by the prover in the first transmission. See, Fiat, A. and Shamir, A., "<u>How to prove yourself: practical solutions to identification and signature problems</u>," Crypto '86, Springer-Verlag (1987), pp. 186-194. References to Fiat/Shamir type signature schemes that are well-known in the art are provided at in the detailed description.) This inventive technique will now be described, and will henceforth be referred to as the "general construction technique."

<u>General Construction Technique.</u>

A triple consisting of a secret key of $\mathcal{U}$, a matching public key of $\mathcal{U}$, and a secret-key certificate on the public key of $\mathcal{U}$, is characterized by (1) the signature scheme that is used by

the CA and (2) the type of key pair of $\mathcal{U}$ that is certified. Denoting the public key of $\mathcal{U}$ by h, and that of the CA by h_0, a secret-certificate on h, when constructed by applying the general construction technique to a Fiat/Shamir type signature scheme, is in effect a signature of this underlying Fiat/Shamir type on the message h, made with a secret key that corresponds, under the signature scheme used by the CA, to public key $h_0 h$.

The general certificate simulation technique for the resulting secret-key certificates consist of generating h as $h_0^{-1} h'_0$, such that one knows a secret key that corresponds to h'_0 under the signature scheme used by the CA. This enables one to generate pairs consisting of a public-key and a secret-key certificate on the public key, without cooperation of the CA.

As will be appreciated, the examples illustrate techniques and concepts of the present invention, but they are only intended to be suggestive and not limiting in any way. For example, other construction techniques than the one described in the preceding paragraphs, or variations thereof, may be used as well. An example of this will be provided at the end of the first preferred embodiment.

FIRST PREFERRED EMBODIMENT

In the first preferred embodiment computations are performed in a (multiplicatively written) group of prime order q, for which efficient algorithms are known to multiply, determine equality of elements, test membership, and to randomly select elements. This group will henceforth be denoted by G_q. No feasible methods should be known to compute discrete logarithms in G_q. Various types of such groups are known. For example, one can take the unique subgroup of order q of $\mathbb{Z}_p^*$, where p is a prime number such that q is a divisor of $p-1$. Another example is an elliptic curve over a finite field. For this reason, no explicit choice for G_q is made in the descriptions.

An expression such as g^x should be understood to be a computation in G_q. In case computations modulo q are performed, (as, for example, in $r_0 = c(x_0 + x) + w_0 \bmod q$), the

modulo operator will be denoted explicitly. In case an element is chosen from a group, it is implicitly assumed that it is the smallest positive representative. The same holds for outcomes of computations.

1. First Exemplary Secret-Key Certificate.

A first exemplary secret-key certificate in the first preferred embodiment, constructed by applying the general construction technique to the Schnorr signature scheme (See, Schnorr, C., "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol. 4, No. 3 (1991), pp. 161-174), will now be described in detail.

Key generation means of the KAC: The secret key of the CA is a number x_0 in $\mathbb{Z}_q$, and the corresponding public key is (g, h_0) in $G_q \times G_q$, where g is a generator of G_q and h_0 denotes g^{x_0}. Preferably, x_0 is chosen uniformly at random in $\mathbb{Z}_q$, and g is chosen uniformly at random in $G_q \setminus \{1\}$.

The pair (g, h_0) and the description of G_q are made publicly known by the CA. The CA also makes publicly known a hash-function $\mathcal{H}$, which maps its arguments to, say, $\mathbb{Z}_{2^t}$, for some appropriate security parameter t (as will be clear to those of ordinary skill in the art, instead of $\mathbb{Z}_{2^t}$, any $\mathbb{Z}_l$ for sufficiently large l can be chosen---this choice is merely for concreteness). This function should meet the requirements that are believed to make the Schnorr signature scheme secure. Preferably, $\mathcal{H}$ is collision-free: this means that it is unfeasible to compute two distinct arguments that are mapped by $\mathcal{H}$ to the same outcome. Functions that are believed to be collision-free are well-known to those of ordinary skill in the art.

Certificate verification means: A secret-key certificate on a public key h in G_q of $\mathcal{U}$ is a pair (c, r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_q$ such that c is equal to $\mathcal{H}(h, g^r(h_0 h)^{-c}, I)$. Here, I is a string containing the name of $\mathcal{U}$, and possibly additional information such as address, employer, electronic mail, and a list of access rights. The incorporation of I in the hash-value is not strictly necessary, but may be required in practice.

The secret-key certificate can alternatively be taken to be

a pair (a, r) in $G_q \times \mathbb{Z}_q$. In that case, the pair is a secret-key certificate on h if $g^r(h_0h)^{-c}$ is equal to a, where c is computed as $\mathcal{H}(h, a, I)$.

As will be clear to those of ordinary skill in the art, the secret-key certificate has been constructed from the Schnorr identification scheme by applying the general construction technique: the certificate is in effect a Schnorr signature on h made with secret key $\log_g(h_0h)$, corresponding to the public key h_0h.

Key generation means of $\mathcal{U}$: In addition to the signature scheme employed by the CA, the type of key pair of $\mathcal{U}$ must be specified in order to define the secret-key certificate. In a general form, the secret key of $\mathcal{U}$ can be taken to be a tuple $(x_1, \ldots, x_k)$, such that each x_i is in $\mathbb{Z}_q$, and h is equal to $\prod_{i=1}^{k} g_i^{x_i}$. Here, $g_1, \ldots, g_k$ are randomly chosen generators of G_q (they need not all be different from g), that are published by the CA in addition to g, h_0, and the descriptions of G_q and $\mathcal{H}$.

For each g_i $(1 \leq i \leq k)$, the CA should preferably know $\log_g g_i$ (in order to be able to conduct the issuing protocol). Hereto, the CA may generate $g_1, \ldots, g_k$ as follows: it generates at random $y_1, \ldots, y_k$ in $\mathbb{Z}_q$, and sets g_i equal to g^{y_i}. (The CA can take one of the numbers g_i equal to g; as will be demonstrated by the flowchart of FIG. 6, such a choice allows restrictive blinding of the issuing protocol. As will be obvious to those of ordinary skill in the art, instead of taking one of the g_i's equal to g, the same effect is obtained by taking this g_i equal to a publicly known power of g.) Observe that the secret key that corresponds to the public key h_0h, in the signature scheme employed by the CA, is the number $\log_g(h_0h)$, which is equal to $x_0 + \sum_{i=1}^{k} y_i x_i \bmod q$; if the described generation process for $g_1, \ldots, g_k$ is used, then the CA can compute this number.

In practice, one may want to use a simpler form of key pair. The simplest form is one in which the secret key of $\mathcal{U}$ is a number x in G_q, and the public key h is equal to g^x (that is, there is only one g_i, and it has been taken equal to g). This enables $\mathcal{U}$ to perform such cryptographic tasks as computing Schnorr signatures and proving knowledge of his secret key (three detailed examples will be provided below). Another

simple form is one in which the secret key of $\mathcal{U}$ is a pair (x_1, x_2), such that h is equal to $g_1^{x_1} g_2^{x_2}$. Here, as mentioned, one of g_1 or g_2 may be taken equal to g (or both, but then there is no advantage in this form over the more efficient form where h is equal to g^x). This form also enables $\mathcal{U}$ to perform cryptographic tasks such as computing signatures and proving knowledge of his secret key. As will be demonstrated in the flowchart of FIG. **6**, this second form is of particular importance for the construction of restrictive blind signature protocols, in that for such a protocol it is required that g_1 is taken equal to g (or a publicly known power of g), and $\mathcal{U}$ should not be able to compute $\log_g g_2$ or $\log_{h_0} g_2$.

Certificate simulating means: Anyone can feasibly generate pairs $h, (c, r)$ such that the verification relation holds, by taking h equal to $h_0^{-1} g^s$ for an arbitrary s in $\mathbb{Z}_q$, and a equal to g^t for an arbitrary t in $\mathbb{Z}_q$; the pair $(c, sc + t \bmod q)$, with c equal to $\mathcal{H}(h, a, I)$, then is a secret-key certificate on h. However, the ability to feasibly generate such pairs together with a secret key, that corresponds to the public key h according to one of the key generation schemes for $\mathcal{U}$ described in the preceding three paragraphs, enables one to forge Schnorr signatures.

A complete secret-key certificate system also requires the description of a certificate issuing protocol, in addition to the description of the secret key certificate itself. A variety of exemplary protocols for issuing the described secret-key certificate can be constructed. Each such certificate issuing protocol can be characterized by the degree by which $\mathcal{U}$, to whom the certificate is issued by the CA, can "randomize" and "blind" the secret key, the public key, and the secret-key certificate. Before providing exemplary embodiments for various certificate issuing protocols, though, a few examples are provided that will help those of ordinary skill in the art to appreciate how the described secret-key certificate may be used in practice.

1.1. Examples for the first exemplary secret-key certificate.

Without loss of generality, it will be assumed in the examples

that a public-key directory is used. (Alternatively, the public keys and secret-key certificates on the public keys may be sent on request.) The entries in the public-key directory will be of the following form:

User	(Public key, Certificate)
I_1	h_1, (c_1, r_1)
I_2	h_2, (c_2, r_2)
$\vdots$	$\vdots$
I_k	h_k, (c_k, r_k)

By virtue of the simulatability of secret-key certificates, any one of the entries could have been generated without cooperation of the CA.

As will be clear to those of ordinary skill in the art, the numbers I_i in the entries of the public-key directory need not refer to the identity of the party associated with the entry: instead, they may refer to a pseudonym of the party, and the party may have a plurality of such pseudonyms.

For simplicity, in each of the examples it will be assumed that the key pair of $\mathcal{U}$ is of the simplest form; the secret key is a number x in $\mathbb{Z}_q$, and the corresponding public key h is equal to g^x. As will be clear to those of ordinary skill in the art, similar examples can be provided when $\mathcal{U}$ uses a different type of key pair. In particular, many techniques and examples in which the more general form of key pair for $\mathcal{U}$ is used, are described and claimed in patent application Ser. No. PCT/NL94/00179.

Example 1: Suppose that user $\mathcal{U}_1$ wants to transfer an encrypted message m in G_q to $\mathcal{U}_2$ (by electronic facsimile, electronic mail, or any other suitable medium). The encryption scheme which is used is the ElGamal scheme (See, ElGamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, Vol. IT-31, No. 4, July 1985, pp. 469--472) in the group G_q. From the public-key directory, $\mathcal{U}_1$ retrieves the public key h_2 of $\mathcal{U}_2$, and the string (c_2, r_2). If c_2 is equal to $\mathcal{H}(h_2, g^{r_2}(h_0 h_2)^{-c_2}, I_2)$, then $\mathcal{U}_1$ can safely transfer the encrypted message to $\mathcal{U}_2$. Hereto, he generates at random a

number s in $\mathbb{Z}_q$, and transfers the pair $(g^s, h_2^s m)$ to $\mathcal{U}_2$. If $\mathcal{U}_2$ can decrypt and retrieve m, then he must know $\log_g h_2$ (as will be clear to those of ordinary skill in the art, this holds under the Diffie-Hellman key exchange assumption in G_q, and for randomly chosen m). This in turn implies that he could not have generated $(h_2, (c_2, r_2))$ by himself. In other words, $\mathcal{U}_1$ can rest assured that $\mathcal{U}_2$ can recover m if and only if the key pair of $\mathcal{U}_2$ has been certified by the CA. Of course, in practice bidirectional communication between $\mathcal{U}_1$ and $\mathcal{U}_2$ will readily reveal to both parties whether the keys of the other party have been certified by the CA.

Example 2: Suppose that $\mathcal{U}_2$ digitally signs a message m for $\mathcal{U}_1$, using the Schnorr signature scheme (alternatively, the ElGamal signature scheme can be used). $\mathcal{U}_1$ receives from $\mathcal{U}_2$ a pair (c, r) such that c is equal to $\mathcal{H}(m, g^r h_2^{-c})$. If the public key h_2 of $\mathcal{U}_2$ is listed in the public-key directory together with a pair (c_2, r_2) for which c_2 is equal to $\mathcal{H}(h_2, g^{r_2}(h_0 h_2)^{-c_2}, I_2)$, then the fact that $\mathcal{U}_2$ can compute this signature also informs $\mathcal{U}_1$ that the key pair of $\mathcal{U}_2$ has been certified by the CA (unless the Schnorr signature scheme can be broken, but in that case the signature scheme of the users itself is also insecure). As will be appreciated, this fact can be verified by anyone and so has legal significance, as it is unfeasible to forge a triple consisting of (1) a pair consisting of a public key and a secret-key certificate on the public key, (2) a message, and (3) a digital signature on the message made with the secret key corresponding to the forged certified public key.

Example 3: Suppose that $\mathcal{U}_2$ wants to prove to $\mathcal{U}_1$ that his entry in the public-key directory has been certified by the CA, without leaving $\mathcal{U}_1$ with a transcript of the proof that can be used to transfer the conviction to others. $\mathcal{U}_2$ hereto proves knowledge of the secret key $\log_g h_2$ to $\mathcal{U}_1$ in a zero-knowledge protocol, which can be done, for example, as follows. $\mathcal{U}_2$ transfers a number a in G_q to $\mathcal{U}_1$, where a is equal to g^w for a randomly chosen w in $\mathbb{Z}_q$. $\mathcal{U}_1$ responds with challenge c randomly chosen from a predetermined small range, say $\{0,1\}$, and transfers it to $\mathcal{U}_2$. $\mathcal{U}_2$ must respond with $c \log_g h_2 + w \bmod q$, denoted by r for further reference. The

correctness of the response can be verified by $\mathcal{U}_1$ by verifying whether $g^r h_2^{-c}$ is equal to a. This three-move interaction is repeated a substantial number of times, and if $\mathcal{U}_2$ can always respond correctly then he must know $\log_g h_2$, thereby proving that the entry in the directory has been certified by the CA. As is well-known in the art, this is a zero-knowledge protocol. The conviction of $\mathcal{U}_1$ cannot be transferred: transcripts of executions of this protocol are feasible to generate together with entries in the public-key directory, with indistinguishable probability distribution.

1.2. First exemplary certificate issuing protocol.

Certificate issuing means: Various certificate issuing protocols for issuing the described secret-key certificate will now be described. As before, for explicitness it will be assumed in each of these protocols that the key pair of $\mathcal{U}$ is of the simplest form; the secret key of $\mathcal{U}$ is a number x in $\mathbb{Z}_q$, and the corresponding public key h is equal to g^x. Those of ordinary skill are believed to be able to straightforwardly apply the inventive techniques to suit the other types of key pairs for $\mathcal{U}$, described previously.

Recall that the secret-key certificate that will be issued to $\mathcal{U}$ by the CA is a pair (c, r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_q$ such that c is equal to $\mathcal{H}(h, g^r(h_0 h)^{-c}, I)$, where h is the public key of $\mathcal{U}$.

Turning now to FIG. **2**, a flowchart of a first secret-key certificate issuing protocol in the first preferred embodiment will now be described in detail.

Box **21** first shows the CA generating a secret key x in $\mathbb{Z}_q$ for use by $\mathcal{U}$. As indicated in the second line, the corresponding public key h of $\mathcal{U}$ is taken equal to g^x. It will be clear to those of ordinary skill in the art that x may alternatively be generated by $\mathcal{U}$ and then communicated to the CA, or $\mathcal{U}$ and the CA can generate it together, for example in such a manner that x is mutually random.

Box **22** first shows the CA generating at random a number w in $\mathbb{Z}_q$. The second line shows the CA computing g^w, which is denoted by a for further reference. The third and fourth lines show the CA computing $\mathcal{H}(h, a, I)$, which is denoted by c,

and $c(x_0 + x) + w \bmod q$, which is denoted by r. The CA then transfers the secret key x and the pair (c, r) to $\mathcal{U}$, as described by the fifth line.

Box **23** first shows $\mathcal{U}$ computing his public key h, by setting it equal to g^x. The second line indicates that $\mathcal{U}$ verifies if c is equal to the hash-value of the triple $(h, g^r(h_0 h)^{-c}, I)$.

Clearly, if the equality holds then the pair (c, r) is a secret-key certificate on the public key h of $\mathcal{U}$, such that $\mathcal{U}$ knows the secret key corresponding to h.

1.3. Second exemplary certificate issuing protocol.

In the certificate issuing protocol of FIG. **2**, the CA must know the secret key of $\mathcal{U}$.

Turning now to FIG. **3**, a flowchart of a secret-key issuing protocol that hides the secret key of $\mathcal{U}$ from the CA, in the first preferred embodiment, will now be described in detail.

Box **31** first shows $\mathcal{U}$ generating at random a number x in $\mathbb{Z}_q$; this will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h by setting it equal to g^x. $\mathcal{U}$ then transfers h to the CA, as indicated by the third line.

Box **32** first shows the CA generating at random a number w in $\mathbb{Z}_q$. The second line shows the CA computing g^w, which is denoted by a for further reference. The third and fourth lines show the CA computing $\mathcal{H}(h, a, I)$, which is denoted by c, and $cx_0 + w \bmod q$, which is denoted by r_0. The fifth line indicates that the CA transfers the pair (c, r_0) to $\mathcal{U}$.

Box **33** first shows $\mathcal{U}$ verifying whether c is equal to the hash-value of the triple $(h, g^{r_0} h_0^{-c}, I)$. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 + cx \bmod q$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a secret-key certificate on the public key h, such that $\mathcal{U}$ knows the secret key corresponding to h.

In this exemplary protocol, the secret key x can be freely chosen by $\mathcal{U}$. It will be clear to those of ordinary skill in the art that the CA can randomize the secret key of $\mathcal{U}$, for instance by using $hg^{x'}$ instead of h in Box **32**. Here, x' is

randomly chosen by the CA from $\mathbb{Z}_q$, and made known to $\mathcal{U}$ only after the CA has received h (in addition, the CA can be requested to transfer a commit on x' to $\mathcal{U}$ before $\mathcal{U}$ reveals h). In that case, the secret key of $\mathcal{U}$ is equal to $x + x' \bmod q$.

1.4. Third exemplary certificate issuing protocol.

Since there exist many secret-key certificates (c, r) on the same public key, the CA may choose a particular one and encode information in it that can be decrypted by insiders.

Turning now to FIG. **4**, a flowchart of a secret-key issuing protocol that hides the secret key from the CA and prevents the subliminal channel, in the first preferred embodiment, will now be described in detail.

Box **41** first shows the CA generating at random a number w_0 in $\mathbb{Z}_q$. The second line shows the CA computing g^{w_0}, which is denoted by a_0 for further reference. The third line indicates that the CA then transfers a_0 to $\mathcal{U}$.

Box **42** first shows $\mathcal{U}$ generating at random a number x in $\mathbb{Z}_q$; this will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h by setting it equal to g^x. The third line shows $\mathcal{U}$ generating at random a number w in $\mathbb{Z}_q$, and the fourth line shows $\mathcal{U}$ computing g^w, which is denoted by a. Finally, as described in the fifth line, $\mathcal{U}$ transfers the pair (h, a) to the CA.

Box **43** first shows the CA computing $\mathcal{H}(h, a_0 a, I)$, which is denoted by c for further reference. The second line shows the CA computing $cx_0 + w_0 \bmod q$, which is denoted by r_0. As described by the third line, the CA then transfers r_0 to $\mathcal{U}$.

Box **44** first shows $\mathcal{U}$ computing c as did the CA in the first line of Box **43**. The second line indicates that $\mathcal{U}$ verifies whether aa_0 is equal to $g^{r_0} h_0^{-c}$. As the third line displays, if this is the case then $\mathcal{U}$ computes $r_0 + cx + w \bmod q$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a randomized secret-key certificate on the public key h (meaning that the CA could not have encoded any information in it), such that $\mathcal{U}$ knows the secret key corresponding to h.

It will be obvious to those of ordinary skill that $\mathcal{U}$ in Box **42** could transfer w to the CA, instead of a. In Box **43** the CA can then compute $cx_0 + w_0 + w \bmod q$ itself, to which $\mathcal{U}$ now must add only cx modulo q.

1.5. Fourth exemplary certificate issuing protocol.

In the issuing protocols described thus far, the CA knows the public key of $\mathcal{U}$. For ordinary public-key directory applications, this is usually by definition the case, since anyone in such applications should be able to associate the public key with $\mathcal{U}$.

In cryptographic mechanisms for transfer of credentials, however, no public-key directories are used; the information representing a credential is only shown by the owner of the credential if he wants to demonstrate possession of his credential to a recipient. A commonly used mechanism is one in which a public key and a public-key certificate on the public key are transferred; the secret key of $\mathcal{U}$ corresponding to his public key enables $\mathcal{U}$ to do additional things such as sign the transfer, or prove possession of additional information (in on-line mechanisms $\mathcal{U}$ might do without needing to know a secret key, and the public key can be a message or a one-way hash thereof). The credential is issued by the CA, and the type of credential that is issued can for instance be denoted by the type of signature that the CA computes. The certificate of the CA proves the validity of the credential to the recipient, and a signature made by $\mathcal{U}$ with his secret key proves that $\mathcal{U}$ willingly transferred the credential to the recipient. If the transfer mechanism is to be privacy-protected, then the CA should not know what the public key and the certificate are, because these are revealed when transferred.

Turning now to FIG. **5**, a flowchart of a fully blinded secret-key issuing protocol in the second preferred embodiment will now be described in detail.

Contrary to the flowcharts of the preceding figures the string I should obviously not be hashed along with h and a if it reveals identifying information about $\mathcal{U}$, such as his name;

otherwise, there would be no point in blinding the certificate issuing protocol. Although the CA may require $\mathcal{U}$ to hash along a string I that, for instance, indicates another party that $\mathcal{U}$ wishes to transfer the received information to, for convenience the string I will henceforth be omitted.

Box **51** first shows the CA generating at random a number w_0 in $\mathbb{Z}_q$. The second line shows the CA computing g^{w_0}, which is denoted by a_0 for further reference. As described by the third line, the CA then transfers a_0 to $\mathcal{U}$.

Box **52** first shows $\mathcal{U}$ generating at random a number x in $\mathbb{Z}_q$; this will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h, by setting it equal to g^x. The third line shows $\mathcal{U}$ generating at random two numbers t, u in $\mathbb{Z}_q$. Using these random numbers, the fourth line shows how $\mathcal{U}$ blinds a_0, by computing $a_0 g^t h_0^u$; this number is denoted by a for further reference. The fifth line shows $\mathcal{U}$ computing $\mathcal{H}(h, a)$, which is denoted by c, and blinding it, as described by the sixth line, to $c + u \bmod q$; this number is denoted by c_0 for further reference. As described by the seventh line, $\mathcal{U}$ then transfers c_0 to the CA.

Box **53** first shows the CA computing $c_0 x_0 + w_0 \bmod q$, which is denoted by r_0 for further reference. As described by the second line, the CA then transfers r_0 to $\mathcal{U}$.

Box **54** first shows $\mathcal{U}$ verifying whether $g^{r_0} h_0^{-c_0}$ is equal to a_0. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 + cx + t \bmod q$, which is denoted by r.

1.6. Fifth exemplary certificate issuing protocol.

The certificate issuing protocols that have been described thus far in effect demonstrate techniques to incorporate various degrees of randomization that can be applied by $\mathcal{U}$.

A particularly valuable randomization is that in which $\mathcal{U}$ can perfectly blind the public key and the secret-key certificate on the public key, such that the CA gets no information about the pair, but cannot fully blind the secret key corresponding to the public key; more specifically, the secret key of $\mathcal{U}$ is a vector of at least two numbers, and $\mathcal{U}$ will not be able to blind a pre-determined non-constant

function of the numbers in the vector. Such an issuing protocol is a restrictive blind signature protocol, as described and claimed in patent application Ser. No. PCT/NL94/00179.

The benefit is of this technique is that, while $\mathcal{U}$ can show a credential (obtained by retrieving a secret key, a matching public key, and a secret-key certificate on the public key, by performing a restrictive blind signature protocol) without enabling traceability by the CA (because the public key and the certificate are fully blinded), appropriate showing protocols (which require $\mathcal{U}$ to perform additional actions with his secret key) can limit the scope of the actions that $\mathcal{U}$ can perform. In patent application Ser. No. PCT/NL94/00179, a wide variety of techniques for transferring credentials, obtained by performing a restrictive blind issuing protocol, are described and claimed.

Turning now to FIG. **6**, a first flowchart of a restrictive blind secret-key certificate issuing protocol for the first preferred embodiment will now be described in detail. This protocol is also described and claimed in patent application Ser. No. PCT/NL94/00179, and, as will be appreciated, is included here (using the present notation) to clearly demonstrate that the protocol is a restrictive blind secret-key certificate issuing protocol.

The key pair of $\mathcal{U}$ must be different from that used until now, because the secret key must be a vector of at least two numbers. For concreteness, the following choice is made: the secret key of $\mathcal{U}$ is a pair (x, I) in $\mathbb{Z}_q \times \mathbb{Z}_q$ such that $g^x g_1^I$ is equal to h. Here, the CA has generated g_1 by generating at random a (secret) number y in $\mathbb{Z}_q$, and setting g_1 equal to g^y. As will be clear to those of ordinary skill in the art, this implies that it is unfeasible for parties other than the CA to compute $\log_g g_1$.

The second number of this pair, I, will be encoded by the CA into the secret key of $\mathcal{U}$ during the certificate issuing protocol. Although $\mathcal{U}$ is unable to modify I, he will be able to generate x by himself uniformly at random in $\mathbb{Z}_q$. Hence, in effect h is generated at random in G_q, independently from I.

As described before, the number I may be related to the identity of $\mathcal{U}$, but can also contain unrelated information such as a credential specification.

Box **61** first shows the CA generating at random a number w_0 in $\mathbb{Z}_q$. The second line shows the CA computing g^{w_0}, which is denoted by a_0 for further reference. As described by the third line, the CA then transfers a_0 to $\mathcal{U}$.

Box **62** first shows $\mathcal{U}$ generating a number x in $\mathbb{Z}_q$; the pair (x, I) will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h, by setting it equal to $g^x g_1^I$. In addition, as displayed in the third line, $\mathcal{U}$ generates two random numbers t, u in $\mathbb{Z}_q$, which will serve to obtain blinded r and c. The fourth line shows $\mathcal{U}$ computing $a_0 g^t (h_0 g_1^I)^u$, which is denoted by a for further reference. As indicated in the fifth line, $\mathcal{U}$ then computes $\mathcal{H}(h, a)$, which is denoted by c. The sixth line specifies $\mathcal{U}$ computing $c + u \bmod q$, which is denoted by c_0. As described by the seventh line, $\mathcal{U}$ then transfers c_0 to the CA.

Box **63** first shows the CA computing $c_0(x_0 + yI) + w_0 \bmod q$, which is denoted by r_0 for further reference. As described by the second line, the CA then transfers r_0 to $\mathcal{U}$.

Box **64** first shows $\mathcal{U}$ verifying whether $g^{r_0}(h_0 g_1^I)^{-c_0}$ is equal to a_0. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 + cx + t \bmod q$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a secret-key certificate on the public key h of $\mathcal{U}$, such that $\mathcal{U}$ knows a secret key corresponding to h. Although $\mathcal{U}$ has perfectly blinded h and (c, r), it is unfeasible for him to completely blind the secret key. That is, the secret key of $\mathcal{U}$ is a pair (x, I') such that $g^x g_1^{I'}$ is equal to h, and if (c, r) is to be a secret-key certificate on h then I' is equal modulo q to the number I that the CA in Box **63** encoded into its response r.

1.7. More than one receiving party.

As will be appreciated, the protocol displayed in FIG. **6** can also be used by the CA to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$ that is substantially under

control of the CA, such that: $\mathcal{U}$ will get to know the public key and the secret-key certificate on the public key; and the secret key corresponding to the public key is shared between $\mathcal{U}$ and $\mathcal{T}$ in such a way that neither of $\mathcal{U}$ and $\mathcal{T}$ can determine it. To this end, the CA initially makes I known to $\mathcal{T}$ but not to $\mathcal{U}$: the CA only informs $\mathcal{U}$ of g_1^I. The protocol displayed in FIG. **6** remains exactly the same, but now $\mathcal{U}$ in a succeeding certificate showing protocol can only compute signatures, or prove knowledge of the secret key, when $\mathcal{T}$ cooperates: $\mathcal{T}$ knows I, and $\mathcal{U}$ knows x, and the certified public key is equal to $g^x g_1^I$. As will be appreciated, $\mathcal{T}$ does not need to participate in the secret-key certificate issuing protocol due to the initial set-up in which the CA only makes I known to $\mathcal{T}$. In patent application Ser. No. PCT/NL94/00179, techniques are detailed and claimed for $\mathcal{T}$ and $\mathcal{U}$ to conduct a succeeding certificate showing protocol such that: I can be computed when $\mathcal{T}$ and $\mathcal{U}$ perform the showing protocol at least twice in response to different challenges; or I can never be computed, no matter how often $\mathcal{U}$ and $\mathcal{T}$ perform the showing protocol.

Other variations of the issuing protocol, for the case the certificate is issued to $\mathcal{U}$ and $\mathcal{T}$ in the manner described in the preceding paragraph, will be obvious to those of ordinary skill in the art. One such variation is that g_1 is taken equal to g in FIG. **6** (and correspondingly y equals 1): in that case, $\mathcal{T}$ at the end of the issuing protocol knows I, and $\mathcal{U}$ knows x, and the certified public key is equal to g^{x+I}. As will be appreciated, the resulting issuing protocol in effect is that described by FIG. **5**, where the role of the CA is now played by the CA and $\mathcal{T}$ together. Instead of using secret key x_0, their secret key now is equal to $x_0 + I \bmod q$.

2. Second Exemplary Secret-Key Certificate.

Each of the exemplary flowcharts that has been described thus far demonstrates a specific type of issuing protocol. The secret-key certificate that is issued is the same in all the flowcharts.

As described earlier, the general construction technique can be applied to any other signature scheme of the Fiat/Shamir

type as well. Although it is believed that the detailed descriptions provided thus far will enable those of ordinary skill in the art to straightforwardly apply the general construction technique to other Fiat/Shamir type signatures, the general construction technique will now be applied to several other Fiat/Shamir type signatures for illustrative purposes. No issuing protocols will be described for these certificates, since it is believed to be an easy matter for those of ordinary skill in the art to apply the inventive techniques of FIGS. **2** to **6** to suit the new certificates (except for one particular certificate, that does not seem to allow blinded issuing; this will be indicated in the description of that particular certificate).

A second exemplary secret-key certificate in the first preferred embodiment, constructed by applying the general construction technique to the Okamoto signature scheme (See, Okamoto, T., Section 6.1. in "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes," Crypto '92, Lecture Notes in Computer Science 740, Springer-Verlag (1993), pp. 31--53), will now be described.

Key generation means of the KAC: The secret key of the CA is a pair (x_1, x_2) in $\mathbb{Z}_q \times \mathbb{Z}_q$, and the corresponding public key is (g_1, g_2, h_0) in $G_q \times G_q$, where g_1 and g_2 are generators of G_q, and h_0 denotes $g_1^{x_1} g_2^{x_2}$. Preferably, x_1 and x_2 are chosen uniformly at random in $\mathbb{Z}_q$, and g_1 and g_2 are chosen uniformly at random from $G_q \setminus \{1\}$.

The tuple (g_1, g_2, h_0) and the description of G_q are made publicly known by the CA. The CA also makes publicly known a hash-function $\mathcal{H}$, which maps its arguments to, say, $\mathbb{Z}_{2^t}$, for some appropriate security parameter t. This function should meet the requirements that are believed to make the Okamoto signature scheme secure. Preferably, $\mathcal{H}$ is collision-free.

Certificate verification means: A secret-key certificate on a public key h in G_q of $\mathcal{U}$ is a triple (c, r_1, r_2) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_q \times \mathbb{Z}_q$ such that c is equal to $\mathcal{H}(h, g_1^{r_1} g_2^{r_2} (h_0 h)^{-c}, I)$.

Alternatively, the secret-key certificate can be taken to be a triple (a, r_1, r_2) in $G_q \times \mathbb{Z}_q \times \mathbb{Z}_q$. In that case, the triple is a secret-key certificate on h if $g_1^{r_1} g_2^{r_2} (h_0 h)^{-c}$ is equal to a, where

c is computed as $\mathcal{H}(h, a, I)$.

The certificate is in effect an Okamoto signature on h made with a secret key that corresponds to the public key $h_0 h$.

Key generation means of $\mathcal{U}$: The discussion of key pairs provided earlier with respect to the first secret-key certificate for the first preferred embodiment, applies here as well. (As will be clear to those of ordinary skill in the art, the symbols g_1, g_2, x_1, x_2 chosen here for convenience, do not refer to the symbols in that discussion.)

Certificate issuing means: Those of ordinary skill in the art are believed to be capable of straightforwardly applying the inventive techniques for (a) the issuing protocols of FIGS. **2** to **6**, and (b) the inventive technique to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$, to construct similar certificate issuing protocols for the present secret-key certificate.

3. Third Exemplary Secret-Key Certificate.

A third exemplary secret-key certificate in the first preferred embodiment, constructed by applying the general construction technique to the Brickell/McCurley signature scheme (See, Brickell, E. and McCurley, K., "An interactive identification scheme based on discrete logarithms and factoring," Journal of Cryptology, Vol. 5, no. 1 (1992), pp. 29-39), will now be described.

As is well-known in the art, the Brickell/McCurley technique can be applied to both the Schnorr and the Okamoto signature scheme. This technique consists of making sure that the order q of the group G_q remains unknown to $\mathcal{U}$; instead, computations that are performed modulo q in the Schnorr or Okamoto scheme are replaced by computations of a large multiple of q. To this end, it will be assumed in the description that G_q is the unique subgroup of order q of $\mathbb{Z}_p^*$, where, p is a prime such that q divides $p-1$, and $p-1$ also contains another prime factor of size comparable to the size of q (such as to prevent efficient factorization of $p-1$). Only the CA may know q, to speed up computations for its internal use. For explicitness,

the application to the Schnorr signature scheme will be assumed.

Key generation means of the KAC: This is the same as for the first secret-key certificate, only this time G_q is of the specific form described above, and q is not made publicly known. In case not even the CA knows q, x_0 is chosen at random from $\mathbb{Z}_{p-1}$.

Certificate verification means: A secret-key certificate on a public key h in G_q of $\mathcal{U}$ is a pair (c,r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_{p-1}$ such that c is equal to $\mathcal{H}(h, g^r(h_0h)^{-c}, I)$.

As with the first secret-key certificate, alternatively a pair (a,r) in $G_q \times \mathbb{Z}_{p-1}$ can be taken to be the certificate.

Key generation means of $\mathcal{U}$: The discussion of key pairs provided earlier with respect to the first secret-key certificate for the first preferred embodiment, applies here as well, with the difference that the secret key (or: each of the numbers in the secret key, for the general form) of $\mathcal{U}$ is chosen in $\mathbb{Z}_{p-1}$.

Certificate issuing means: Again, it is an easy matter to apply the inventive techniques for (a) the issuing protocols of FIGS. **2** to **6**, and (b) the inventive technique to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$, to construct similar certificate issuing protocols for the present secret-key certificate. Hereto, all operations that are performed modulo q must be replaced by operations modulo $p-1$ (if the CA knows q, it can of course still compute g^w, for w in $\mathbb{Z}_{p-1}$, by computing $g^{w \bmod q}$).

4. Fourth Exemplary Secret-Key Certificate.

A fourth exemplary secret-key certificate in the first preferred embodiment, constructed by applying the general construction technique to the DSA (See, NIST, "Specifications for a digital signature standard (DSS)," Federal Information Processing Standards Pub. (draft), Aug. 19, 1991), will now be described.

Key generation means of the KAC: The secret key of the CA is a number x_0 in $\mathbb{Z}_q$, and the corresponding public key is (g,h_0) in $G_q \times G_q$, where g is a generator of G_q and h_0 denotes g^{x_0}.

The pair (g, h_0) and the description of G_q are made publicly known by the CA. The CA also makes publicly known a hash-function $\mathcal{H}$, which maps its arguments to, say, $\mathbb{Z}_{2^t}$, for some appropriate security parameter t. This function should meet the requirements that are believed to make the DSA secure.

<u>Certificate verification means:</u> A secret-key certificate on a public key h in G_q of $\mathcal{U}$ is a pair (a, r) in $\mathbb{Z}_q \times \mathbb{Z}_q$ such that $((g^{cr^{-1}}(h_0 h)^{ar^{-1}}) \bmod q)$ is equal to a, where c denotes $\mathcal{H}(h, I)$.

<u>Key generation means of $\mathcal{U}$:</u> The discussion of key pairs provided earlier with respect to the first secret-key certificate for the first preferred embodiment, applies here as well.

<u>Certificate issuing means:</u> Those of ordinary skill in the art are believed to be capable of straightforwardly modifying the issuing protocol of FIG. **2** to construct a similar certificate issuing protocol for the present secret-key certificate. It is important to note that for this particular realization, contrary to the other secret-key certificates systems described in this application, it is unclear how to construct issuing protocols similar to the issuing protocols of FIGS. **3** to **6**.

<u>5. Fifth Exemplary Secret-Key Certificate.</u>

It will now be demonstrated that certain variations of the general construction technique can be used as well. A fifth exemplary secret-key certificate in the first preferred embodiment, constructed by applying a variation of the general construction technique to the Schnorr signature scheme, will now be described in detail.

<u>Key generation means of the KAC:</u> This is the same as in the description of the first secret-key certificate.

<u>Certificate verification means:</u> A secret-key certificate on a public key h in G_q of $\mathcal{U}$ will now be taken to be a pair (c, r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_q$ such that c is equal to $\mathcal{H}(h, g^r h^{-c}, I)$.

The secret-key certificate can alternatively be taken to be a pair (a, r) in $G_q \times \mathbb{Z}_q$ such that $g^r h^{-c}$ is equal to a, where c is computed as $\mathcal{H}(h, a, I)$.

It seems that this secret-key certificate has not been constructed from the Schnorr identification scheme by applying the general construction technique: the public key h_0 seems to have "disappeared" from the verification relation. However, it now has to show up in the definition of the type of key pair used by $\mathcal{U}$. In other words, merely a slight variation of the general construction technique has been applied.

Key generation means of $\mathcal{U}$: In general, for the modified secret-key certificate to be secure, the public key of $\mathcal{U}$ must be defined as a product $g_1^{x_1} \cdots g_k^{x_k} h_0^{x_{k+1}}$, where none of the randomly chosen elements g_i is equal to g (or a publicly known power thereof). As with the first secret-key certificate, $g_1, \ldots, g_k$ are randomly chosen generators of G_q that are published by the CA in addition to g, h_0, and the descriptions of G_q and $\mathcal{H}$.

In practice, one may want to use a simpler form of key pair. The simplest form is one in which the secret key of $\mathcal{U}$ is a number x in G_q, and the public key h is equal to $(h_0)^x$, and h may not be equal to 1. Another simple form is one in which the secret key of $\mathcal{U}$ is a pair (x_1, x_2), such that h is equal to $g_1^{x_1} h_0^{x_1}$.

To demonstrate that all the issuing techniques provided for the first secret-key certificate in the first preferred embodiment can be straightforward applied to construct issuing protocols for the modified secret-key certificate, the most difficult to realize type of issuing protocol for the modified secret-key certificate will now be described.

Turning now to FIG. 7, a second flowchart of a restrictive blinded secret-key certificate issuing protocol for the first preferred embodiment will now be described in detail.

The secret key of $\mathcal{U}$ is a pair (x, I) in $\mathbb{Z}_q \times \mathbb{Z}_q$, and the public key h is equal to $g_1^x h_0^I$. As in the previous flowchart, the CA has generated g_1 by generating at random a (secret) number y in $\mathbb{Z}_q$, and setting g_1 equal to g^y. The blinding that can be performed by $\mathcal{U}$ in the protocol differs from that in the preceding protocol, but the restrictivity property still holds: the number that the CA will encode into the secret key

(x, I) of $\mathcal{U}$ is equal to $I/x \bmod q$.

As in the preceding flowchart, at the start of the issuing protocol the CA decides on a number I in $\mathbb{Z}_q$ that will be encoded into the secret key of $\mathcal{U}$ when the issuing protocol is performed.

Box **71** first shows the CA generating at random a number w_0 in $\mathbb{Z}_q$. The second line shows the CA computing g^{w_0}, which is denoted by a_0 for further reference. As described by the third line, it then transfers a_0 to $\mathcal{U}$.

Box **72** first shows $\mathcal{U}$ generating a number x in $\mathbb{Z}_q$; the pair $(x, Ix \bmod q)$ will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h, by setting it equal to $(h_0 g_1^I)^x$. In addition, as specified by the third line, $\mathcal{U}$ generates two random numbers t, u in $\mathbb{Z}_q$, which will serve to obtain blinded r and c. The fourth line shows $\mathcal{U}$ computing $a_0^x g^t (h_0 g_1^I)^{xu}$, which is denoted by a. The fifth line shows $\mathcal{U}$ computing $\mathcal{H}(h, a)$, which is denoted by c. The sixth line shows $\mathcal{U}$ computing $c + u \bmod q$, which is denoted by c_0. As described by the seventh line, $\mathcal{U}$ then transfers c_0 to the CA.

Box **73** first shows the CA computing $c_0(x_0 + yI) + w_0 \bmod q$, which is denoted by r_0 for further reference. As described by the second line, the CA then transfers r_0 to $\mathcal{U}$.

Box **74** first shows $\mathcal{U}$ verifying whether $g^{r_0}(h_0 g_1^I)^{-c_0}$ is equal to a_0. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 x + t \bmod q$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a secret-key certificate on the public key h of $\mathcal{U}$, such that $\mathcal{U}$ knows a secret key corresponding to h. The following holds: the secret key of $\mathcal{U}$ is a pair (x, I') such that $h_0^x g_1^{I'}$ is equal to h, and if (c, r) is to be a secret-key certificate on h then $I'/x \bmod q$ is equal modulo q to the number I that the CA in Box **73** encoded into its response r.

SECOND PREFERRED EMBODIMENT

In the second preferred embodiment computations are performed in a multiplicative group modulo n, denoted by $\mathbb{Z}_n^*$, with n being the product of two distinct large primes. Since the order of the group may only be known to at most the certifying

party, the computations in the exponents are modulo a number v that is not a proper divisor of the order of $\mathbb{Z}_n^*$. For this reason, in the blinded certificate issuing protocols that will be described, expressions involving $\operatorname{div} v$ will show up (recall that x is equal to $x \bmod v + v(x \operatorname{div} v)$ for x in $\mathbb{N}$).

Since multiplications and divisions in $\mathbb{Z}_n^*$ are always performed modulo n, the operator $\bmod n$ will never be mentioned explicitly. So for example w^v stands for $w^v \bmod n$. In case other modulo operations are involved, the modulo operator is explicitly mentioned (as in, for example, $c_0 = c + u \bmod v$). If numbers are chosen from a group or ring, always the smallest positive remainder is implied. For instance, $w \in_{\mathcal{R}} \mathbb{Z}_n^*$ implies that w is chosen at random from the subset $\{1, \ldots, n-1\}$ containing the numbers co-prime with n (in practice, this set can be taken to be $\{1, \ldots, n-1\}$).

As will be obvious to those of ordinary skill in the art, one can take v to be either composite or prime.

The exemplary secret-key certificates will, as in the first preferred embodiment, all be constructed from Fiat/Shamir signature schemes by applying the general construction technique. The structure of the exposition is similar to the exposition in the first preferred embodiment: a detailed description of one particular system is presented first, together with a variety of issuing protocols, followed by a description of how the general construction technique can be applied to other Fiat/Shamir type schemes.

1. First Exemplary Secret-Key Certificate.

A first exemplary secret-key certificate in the second preferred embodiment, constructed by applying the general construction technique to the Guillou/Quisquater signature scheme (See, Guillou, L. and Quisquater, J., "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," Lecture Notes in Computer Science 330, Proceedings of Eurocrypt '88, Springer-Verlag (1989), pp. 123-128), will now be described in detail.

Key generation means of the KAC: Let v be an element in $\mathbb{Z}_n^*$.

A convenient choice for v is to take v a prime that is co-prime to $\varphi(n)$ (the number of elements of $\mathbb{Z}_n^*$).

The secret key of the CA is a number x_0 in $\mathbb{Z}_n^*$, and its corresponding public key h_0 is equal to x_0^v. Preferably, x_0 is chosen uniformly at random in $\mathbb{Z}_n^*$.

The triple (n, v, h_0) is made publicly known by the CA. In addition, a hash-function $\mathcal{H}$ is made publicly known, which maps its arguments to, say, $\mathbb{Z}_{2^t}$, for some appropriate security parameter t. This function should meet the requirements that are believed to make the Guillou/Quisquater signature scheme secure, and is preferably collision-free.

Certificate verification means: A secret-key certificate on a public key h in $\mathbb{Z}_n^*$ of $\mathcal{U}$ is a pair (c, r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_n^*$ such that c is equal to $\mathcal{H}(h, r^v(h_0h)^{-c}, I)$. The number I is as described in the first preferred embodiment, and as before its incorporation is not strictly necessary.

The secret-key certificate can alternatively be taken to be a pair (a, r) in $\mathbb{Z}_n^* \times \mathbb{Z}_n^*$. In that case, the pair is a secret-key certificate on h if $r^v(h_0h)^{-c}$ is equal to a, where c is computed as $\mathcal{H}(h, a, I)$.

As will be clear to those of ordinary skill in the art, this secret-key certificate system has been constructed from the Guillou/Quisquater identification scheme by applying the general construction technique. That is, the certificate is in effect a Guillou-Quisquater signature on h made with secret key $(h_0h)^{1/v}$, corresponding to the public key h_0h.

Key generation means of $\mathcal{U}$: In a general form, the secret key of $\mathcal{U}$ can be taken to be a tuple $(z_1, \ldots, z_k; x)$, where each z_i is in $\mathbb{Z}_v$, x is in $\mathbb{Z}_n^*$, and h is equal to $(\prod_{i=1}^k g_i^{z_i})x^{v_1}$. (As will be clear to those of ordinary skill in the art, the more general form, $(z_1, \ldots, z_k; x_1, \ldots, x_l)$, can be considered, such that h is equal to $\prod_{i=1}^k g_i^{z_i} \prod_{i=1}^l x_i^{v_i}$ for appropriate exponents v_i. It is believed to be straightforward for those of ordinary skill in the art to apply the disclosed inventive techniques to this more general form.) Here, $g_1, \ldots, g_k$ are randomly chosen numbers in $\mathbb{Z}_n^*$, preferably of large order (which can easily be taken care of by taking the prime divisors p and q of n such that $p-1$ and $q-1$ both have a large prime divisor). All these

numbers are made publicly available by the CA, in addition to n, v, h_0 and the description of $\mathcal{H}$. As will be appreciated, any integer v_1 can be used for the key pair of $\mathcal{U}$. From now on, for concreteness v_1 will always be taken equal to v.

For each g_i $(1 \leq i \leq k)$, the CA should preferably know $g_i^{1/v}$, in order to be able to conduct the issuing protocol: the secret key of the CA that corresponds to the public key $h_t h$ is the v-th root of this number, which is equal to $x_0 x \prod_{i=1}^{k}(g_i^{1/v})^{x_i}$.

In practice, one may want to use a simpler form. The simplest form is one in which the secret key of $\mathcal{U}$ is a number x in Z_n^*, and the public key h is equal to x^v. Another simple form is one in which the secret key of $\mathcal{U}$ is a pair $(x_1; x_2)$ in $\mathbb{Z}_v \times \mathbb{Z}_n^*$, such that h is equal to $g_1^{x_1} x_2^v$. Both forms allow $\mathcal{U}$ to, for instance, compute signatures and prove knowledge of his secret key. (As is well-known, for a key pair of the form $g_1^{x_1} g_2^{x_2}$ no secure protocols for these tasks are known in the art.) As will be demonstrated in the flowchart of FIG. **12**, this second form is of particular importance for the construction of restrictive blind signature protocols.

Certificate simulating means: Those of ordinary skill in the art may wish to verify that anyone can feasibly generate pairs $h, (c, r)$ such that the verification relation holds, by taking h equal to $h_0^{-1} s^v$ for an arbitrary s in $\mathbb{Z}_n^*$, and a equal to t^v for an arbitrary t in $\mathbb{Z}_n^*$; the pair $(c, s^c t)$, with c equal to $\mathcal{H}(h, a, I)$, then is a secret-key certificate on h. However, the ability to feasibly generate such pairs together with a secret key, that corresponds to the public key h according to one of the key schemes described in the preceding three paragraphs, enables one to forge Guillou/Quisquater signatures.

Before providing exemplary embodiments for various certificate issuing protocols, a few examples are provided that will help those of ordinary skill in the art to appreciate how the described secret-key certificate may be used in practice.

1.1. Examples for the first exemplary secret-key certificate.

Example 1: In this example, it will be assumed that the secret key of $\mathcal{U}$ is a number x in $\mathbb{Z}_v$, and the corresponding public key

h is equal to g^x. Here, g is a randomly chosen number in $\mathbb{Z}_n^*$, preferably of large order, which has been made publicly available by the CA. Suppose that user $\mathcal{U}_1$ wants to transfer an encrypted message m in $\mathbb{Z}_n^*$ to $\mathcal{U}_2$. The encryption scheme which is used is the ElGamal scheme in the group $\mathbb{Z}_n^*$. From the public-key directory, $\mathcal{U}_1$ retrieves the public key h_2 of $\mathcal{U}_2$, and the string (c_2, r_2). If c_2 is equal to $\mathcal{H}(h_2, r_2^v(h_0h_2)^{-c_2}, I_2)$, then $\mathcal{U}_1$ can safely transfer the encrypted message to $\mathcal{U}_2$. Hereto, he generates at random a number s, and transfers the pair $(g^s, h_2^s m)$ to $\mathcal{U}_2$. If $\mathcal{U}_2$ can decrypt and retrieve m, then he must know $\log_g h_2$ (as is well-known in the art, this holds under the factoring assumption, and for randomly chosen m). This in turn implies that he could not have generated $(h_2, (c_2, r_2))$ by himself.

<u>Example 2:</u> In this example, it will be assumed that the secret key of $\mathcal{U}$ is a number x in $\mathbb{Z}_n^*$, and the corresponding public key h is equal to x^v. Suppose that $\mathcal{U}_2$ digitally signs a message m for $\mathcal{U}_1$, using the Guillou/Quisquater signature scheme. $\mathcal{U}_1$ receives from $\mathcal{U}_2$ a pair (c, r) such that c is equal to $\mathcal{H}(m, r^v h_2^{-c})$. If the public key h_2 of $\mathcal{U}_2$ is listed in the public-key directory together with a pair (c_2, r_2) for which c_2 is equal to $\mathcal{H}(h_2, r_2^v(h_0h_2)^{-c_2}, I_2)$, then the fact that $\mathcal{U}_2$ can compute this signature also informs $\mathcal{U}_1$ that the key pair of $\mathcal{U}_2$ has been certified by the CA. These two facts can also be verified by anyone else, and so the signature can have legal significance.

<u>1.2. First exemplary certificate issuing protocol.</u>

A variety of protocols for issuing the described secret-key certificate, similar to those described in the first preferred embodiment, will now be described. For explicitness it will be assumed in each of these flowcharts that the secret key of $\mathcal{U}$ is of the simplest form; it is a number x in $\mathbb{Z}_n^*$ such that the corresponding public key h is equal to x^v. Those of ordinary skill are believed to be able to straightforwardly apply the inventive techniques to suit the other types of key pairs for $\mathcal{U}$.

Turning now to FIG. 8, the first flowchart of a secret-key

certificate issuing protocol for the second preferred embodiment will now be described in detail. As will be clear to those of ordinary skill in the art, this issuing protocol has the same functionality as the protocol of FIG. **2**.

Box **81** first shows the CA generating a secret key x in $\mathbb{Z}_n^*$ for use by $\mathcal{U}$. As indicated in the second line, the corresponding public key h of $\mathcal{U}$ is taken equal to x^v.

Box **82** first shows the CA generating at random a number w in $\mathbb{Z}_n^*$. The second line shows the CA computing w^v, which is denoted by a for further reference. The third and fourth lines show the CA computing $\mathcal{H}(h, a, I)$, which is denoted by c, and $(x_0 x)^c w$, which is denoted by r. The CA then transfers the secret key x and the pair (c, r) to $\mathcal{U}$, as described by the fifth line.

Box **83** first shows $\mathcal{U}$ computing his public key h, by setting it equal to x^v. The second line indicates that $\mathcal{U}$ verifies if c is equal to the hash-value of the triple $(h, r^v(h_0 h)^{-c}, I)$.

If the equality holds, the pair (c, r) is a secret-key certificate on the public key h of $\mathcal{U}$, such that $\mathcal{U}$ knows the secret key corresponding to h.

As will be clear to those of ordinary skill in the art, if the CA knows the prime factorization of n, then in Box **82** it can simply take any number a in $\mathbb{Z}_n^*$. Of course, since v-th roots are involved, the CA must generate this number from the set of all v-th powers in $\mathbb{Z}_n^*$; if v is co-prime to $\varphi(n)$ then this set is equal to $\mathbb{Z}_n^*$; if, for example, v is twice a number that is co-prime to $\varphi(n)$, then v-th roots modulo n exist only for the quadratic residues in $\mathbb{Z}_n^*$.

1.3. Second exemplary certificate issuing protocol.

As in the first preferred embodiment, the need for the CA to know the secret key of $\mathcal{U}$ can be removed by letting $\mathcal{U}$ perform part of the computations.

Turning now to FIG. **9**, a flowchart of a secret-key issuing protocol that hides the secret key of $\mathcal{U}$ from the CA, in the second preferred embodiment, will now be described in detail. As will be clear to those of ordinary skill in the art, this issuing protocol has the same functionality as the protocol of

FIG. 3.

Box **91** first shows $\mathcal{U}$ generating at random a number x in $\mathbb{Z}_n^*$; this will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h by setting it equal to x^v. $\mathcal{U}$ then transfers h to the CA, as indicated by the third line.

Box **92** first shows the CA generating at random a number w in $\mathbb{Z}_n^*$. The second line shows the CA computing w^v, which is denoted by a for further reference. The third and fourth lines show the CA computing $\mathcal{H}(h,a,I)$, which is denoted by c, and $x_0^c w$, which is denoted by r_0. The fifth line indicates that the CA transfers the pair (c,r_0) to $\mathcal{U}$.

Box **93** first shows $\mathcal{U}$ verifying whether c is equal to the hash-value of the triple $(h, r_0^v h_0^{-c}, I)$. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 x^c$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c,r) is a secret-key certificate on the public key h, such that $\mathcal{U}$ knows the secret key corresponding to h.

1.4. Hiding the secret key from the CA.

Contrary to the first preferred embodiment, the RSA function has a trapdoor (i.e., the prime factorization of n). If the CA knows this prime factorization, it can always compute the secret key, even if $\mathcal{U}$ tries to hide it by using the issuing protocol of FIG. **9**. Nevertheless, it can still make sense for $\mathcal{U}$ to hide the secret key from the CA, namely, in the case multiple secret keys correspond to the same public key and $\mathcal{U}$ can know only one (or a small fraction of all corresponding secret keys). The following example may help to appreciate this.

Consider a situation where the secret key of $\mathcal{U}$ is a tuple $(x_1, x_2; I_1, I_2)$ in $\mathbb{Z}_n^* \times \mathbb{Z}_n^* \times \mathbb{Z}_v \times \mathbb{Z}_v$, such that h_1 is equal to $g_1^{I_1} x_1^v$ and h_2 is equal to $g_1^{I_2} x_2^v$, where (h_1, h_2) is his public key. As is well-known in the art, this public key can be used to make a one-time signature. The signature of $\mathcal{U}$ on a message m in $\mathbb{Z}_v$ is the pair $(I_1 m + I_2 \bmod v, g_1^{I_1 m + I_2 \operatorname{div} v} x_1^m x_2)$. A straightforward modification of the preceding certificate issuing protocol

gets $\mathcal{U}$ a secret-key certificate on his public key (h_1, h_2). (The certificate is a pair (c, r) such that c is equal to $(h_1, h_2, r^v(h_0 h_1)^{-c}, I)$, for instance.) Of course, as will be clear to those of ordinary skill in the art, to prevent forgery of signatures, the number m in the verification relation for the signature now must be taken equal to a one-way hash of at least the message and the public key, instead of being the message itself. Now, even if the CA knows the prime factorization, and hence can compute all secret keys corresponding to (h_1, h_2), the probability is negligible that it can determine the particular secret key known by $\mathcal{U}$; as is well-known in the art, the signature scheme of $\mathcal{U}$ is witness-indistinguishable. This in turn implies that if the CA forges a signature of $\mathcal{U}$, using a different secret key, then the $\mathcal{U}$ can compute $g_1^{1/v}$, thereby proving the fraud of the CA.

If the CA should not have the power to compute any secret key at all, the number n must be generated such that the CA does not know the prime factors. To this end, the process of generating n should be conducted by a trusted secured device that destroys the prime factors after having generated n, or by some other trusted party.

1.5. Third exemplary certificate issuing protocol.

As in the first preferred embodiment, there exist many secret-key certificates (c, r) on the same public key and so the CA may choose a particular one and encode information in it.

Turning now to FIG. **10**, a flowchart of a secret-key issuing protocol that hides the secret key from the CA and prevents the subliminal channel, in the second preferred embodiment, will now be described in detail. As will be clear to those of ordinary skill in the art, this issuing protocol has the same functionality as the protocol of FIG. **4**.

Box **101** first shows the CA generating at random a number w_0 in $\mathbb{Z}_n^*$. The second line shows the CA computing w_0^v, which is denoted by a_0 for further reference. The third line indicates that the CA then transfers a_0 to $\mathcal{U}$.

Box **102** first shows $\mathcal{U}$ generating at random a number x in $\mathbb{Z}_n^*$; this will be his secret key. The second line shows $\mathcal{U}$

computing the corresponding public key h by setting it equal to x^v. The third line shows $\mathcal{U}$ generating at random a number w in $\mathbb{Z}_n^*$, and the fourth line shows $\mathcal{U}$ computing w^v, which is denoted by a. Finally, as described in the fifth line, $\mathcal{U}$ transfers the pair (h, a) to the CA.

Box **103** first shows the CA computing $\mathcal{H}(h, a_0 a, I)$, which is denoted by c for further reference. The second line shows the CA computing $x_0^c w_0$, which is denoted by r_0. As described by the third line, the CA then transfers r_0 to $\mathcal{U}$.

Box **104** first shows $\mathcal{U}$ computing c as did the CA in the first line of Box **103**. The second line indicates that $\mathcal{U}$ verifies whether aa_0 is equal to $r_0^v h_0^{-c}$. As the third line displays, if this is the case then $\mathcal{U}$ computes $r_0 x^c w$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a secret-key certificate on the public key h, randomized by $\mathcal{U}$, such that $\mathcal{U}$ knows the secret key corresponding to h.

It will be obvious to those of ordinary skill that $\mathcal{U}$ in Box **102** could transfer w to the CA, instead of a. In Box **103** the CA can then compute $x_0^c w_0 w$ itself, into which $\mathcal{U}$ now has to multiply x^c modulo n. This, however, causes an extra computational cost for the CA, which now has to compute one additional exponentiation in $\mathbb{Z}_n^*$, whereas the computational cost for $\mathcal{U}$ is virtually not reduced.

1.6. Fourth exemplary certificate issuing protocol.

Turning now to FIG. **11**, a flowchart of a fully blinded secret-key issuing protocol in the second preferred embodiment will now be described in detail. As will be clear to those of ordinary skill in the art, this issuing protocol has the same functionality as the protocol of FIG. **5**.

As discussed in the first preferred embodiment, for convenience the string I will henceforth be omitted.

Box **111** first shows the CA generating at random a number w_0 in $\mathbb{Z}_n^*$. The second line shows the CA computing w_0^v, which is denoted by a_0 for further reference. As described by the third line, the CA then transfers a_0 to $\mathcal{U}$.

Box **112** first shows $\mathcal{U}$ generating at random a number x in

$\mathbb{Z}_n^*$; this will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h, by setting it equal to x^v. The third line shows $\mathcal{U}$ generating at random a number t in $\mathbb{Z}_n^*$, and the fourth line shows $\mathcal{U}$ generating at random a number u in $\mathbb{Z}_v$. Using these random numbers, the fifth line shows how $\mathcal{U}$ blinds a_0, by computing $a_0 t^v h_0^u$; this number is denoted by a for further reference. The sixth line shows $\mathcal{U}$ computing $\mathcal{H}(h,a)$, which is denoted by c, and blinding it, as described by the seventh line, to $c+u \bmod v$; this number is denoted by c_0 for further reference. As described by line eight, $\mathcal{U}$ then transfers c_0 to the CA.

Box **113** first shows the CA computing $x_0^{c_0} w_0$, which is denoted by r_0 for further reference. As described by the second line, the CA then transfers r_0 to $\mathcal{U}$.

Box **114** first shows $\mathcal{U}$ verifying whether $r_0^v h_0^{-c_0}$ is equal to a_0. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 x^c t h_0^{c+u \operatorname{div} v}$, which is denoted by r.

As will be clear to those of ordinary skill in the art, (c,r) is a secret-key certificate on the public key h of $\mathcal{U}$. In addition, views of the CA in executions of this issuing protocol are independent from pairs $(h,(c,r))$.

1.7. Fifth exemplary certificate issuing protocol.

Turning now to FIG. **12**, a flowchart of a restrictive blind secret-key certificate issuing protocol for the second preferred embodiment will now be described in detail. This protocol is also described and claimed in patent application Ser. No. PCT/NL94/00179, but, as will be appreciated, is included here (using the present notation) to clearly demonstrate that the protocol in effect is a restrictive blind secret-key certificate issuing protocol. As will be clear to those of ordinary skill in the art, this issuing protocol has the same functionality as the protocol of FIG. **6**.

The key pair of $\mathcal{U}$ must be different from that used until now, because the secret key must be a vector of at least two numbers. For concreteness, the following choice is made: the secret key of $\mathcal{U}$ is a pair $(x;I)$ in $\mathbb{Z}_n^* \times \mathbb{Z}_v$ such that $x^v g_1^I$ is equal to h. Here, the CA has generated g_1 by generating at

random a (secret) number y in $\mathbb{Z}_n^*$, and setting g_1 equal to y^v.

The second number of this pair, I, will be encoded by the CA into the secret key of $\mathcal{U}$ during the certificate issuing protocol, in such a way that $\mathcal{U}$ will not be able to change I to a number I' that differs modulo v from I. On the other hand, $\mathcal{U}$ will be able to generate x by himself uniformly at random in $\mathbb{Z}_n^*$, and hence h in effect is generated at random from $\mathbb{Z}_n^*$, independently from I. As described before, the number I may be related to the identity of $\mathcal{U}$, but can as well contain unrelated information, such as a credential specification.

Box **121** first shows the CA generating at random a number w_0 in $\mathbb{Z}_n^*$. The second line shows the CA computing w_0^v, which is denoted by a_0 for further reference. As described by the third line, the CA then transfers a_0 to $\mathcal{U}$.

Box **122** first shows $\mathcal{U}$ generating a number x in $\mathbb{Z}_n^*$; the pair (x, I) will be his secret key. The second line shows $\mathcal{U}$ computing the corresponding public key h, by setting it equal to $x^v g_1^I$. In addition, as displayed in the third and fourth lines, $\mathcal{U}$ generates two random numbers t in $\mathbb{Z}_n^*$ and u in $\mathbb{Z}_v$, which will serve to obtain blinded r and c. The fifth line shows $\mathcal{U}$ computing $a_0 t^v (h_0 g_1^I)^u$, which is denoted by a for further reference. As indicated in the sixth line, $\mathcal{U}$ then computes $\mathcal{H}(h, a)$, which is denoted by c. The seventh line specifies $\mathcal{U}$ computing $c + u \bmod v$, which is denoted by c_0. As described by the eighth line, $\mathcal{U}$ then transfers c_0 to the CA.

Box **123** first shows the CA computing $(x_0 y^I)^{c_0} w_0$, which is denoted by r_0 for further reference. As described by the second line, the CA then transfers r_0 to $\mathcal{U}$.

Box **124** first shows $\mathcal{U}$ verifying whether $r_0^v (h_0 g_1^I)^{-c_0}$ is equal to a_0. As described by the second line, if this is the case then $\mathcal{U}$ computes $r_0 x^c t (h_0 g_1^I)^{c+u \operatorname{div} v}$, which is denoted by r.

As can easily be verified by those of ordinary skill in the art, the pair (c, r) is a secret-key certificate on the public key h of $\mathcal{U}$, such that $\mathcal{U}$ knows the secret key corresponding to h. Although $\mathcal{U}$ has perfectly blinded h and (c, r), it is unfeasible for him to completely blind the secret key. The secret key of $\mathcal{U}$ is a pair $(x; I')$ such that $x^v g_1^{I'}$ is equal to h, and if (c, r) is to be a secret-key certificate on h then I' is

unavoidably equal modulo v to the number I that the CA in Box **123** encoded into its response r.

1.8. More than one receiving party.

As in the first preferred embodiment, the protocols displayed in FIGS. **11** and **12** can also be used by the CA to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$ that is substantially under control of the CA, such that: $\mathcal{U}$ will get to know the public key and the secret-key certificate on the public key; and the secret key corresponding to the public key is shared between $\mathcal{U}$ and $\mathcal{T}$ in such a way that neither of $\mathcal{U}$ and $\mathcal{T}$ can determine it.

One possible such use, based on the protocol of FIG. **11**, will now be described. The secret key used that will be used for $\mathcal{U}$ by the CA now is Ix_0 (the public key still is x_0^v), where I is known by $\mathcal{T}$ but not by $\mathcal{U}$; $\mathcal{U}$ only knows I^v. The issuing protocol between the CA and $\mathcal{U}$ as described by FIG. **11** now takes place, where the CA now computes in the first line of Box **113** the number r_0 as $(x_0I)^{c_0}w_0$. As will be clear to those of ordinary skill in the art, $\mathcal{T}$ at the end of the issuing protocol knows I, and $\mathcal{U}$ knows x, and the certified public key is equal to $(Ix)^v$. As will be appreciated, $\mathcal{T}$ does not need to participate in the secret-key certificate issuing protocol due to the initial set-up in which the CA only makes I known to $\mathcal{T}$. In patent application Ser. No. PCT/NL94/00179, techniques are detailed and claimed for $\mathcal{T}$ and $\mathcal{U}$ to conduct a certificate showing protocol following the issuing protocol.

Other variations of the issuing protocol, for the case the certificate is issued to $\mathcal{U}$ and $\mathcal{T}$ in the manner described in the preceding paragraph (such as a variation in which $\mathcal{T}$ and $\mathcal{U}$ end up with a certified public key of the form $g_1^I x^v$), are believed to be obvious to construct for those of ordinary skill in the art.

2. Second Exemplary Secret-Key Certificate.

As in the first preferred embodiment, a variety of other secret-key certificates will now be described, each of which is constructed from a Fiat/Shamir type signature scheme by applying the general construction technique.

A second exemplary secret-key certificate in the second preferred embodiment, constructed by applying the general construction technique to the Okamoto signature scheme (See, Okamoto, T., Section 3.2./3.3. and Section 6 in "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes," Crypto '92, Lecture Notes in Computer Science 740, Springer-Verlag (1993), pp. 31--53), will now be described.

Key generation means of the KAC: The secret key of the CA is a pair $(x_1; x_2)$ in $\mathbb{Z}_v \times \mathbb{Z}_n^*$, and the corresponding public key h_0 is equal to $g^{x_1}x_2^v$, where g is an element of large order in $\mathbb{Z}_n^*$.

The tuple (g, v, h_0, n) is made publicly known by the CA. The CA also makes publicly known a hash-function $\mathcal{H}$, which maps its arguments to, say, $\mathbb{Z}_{2^t}$, for some appropriate security parameter t. This function should meet the requirements that are believed to make the Okamoto signature scheme secure.

Certificate verification means: A secret-key certificate on a public key h in $\mathbb{Z}_n^*$ of $\mathcal{U}$ is a triple (c, r_1, r_2) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_v \times \mathbb{Z}_n^*$ such that c is equal to $\mathcal{H}(h, g^{r_1}r_2^v(h_0h)^{-c}, I)$.

Alternatively, the secret-key certificate can be taken to be a triple (a, r_1, r_2) in $\mathbb{Z}_n^* \times \mathbb{Z}_n^* \times \mathbb{Z}_n^*$. In that case, the triple is a secret-key certificate on h if $g^{r_1}r_2^v(h_0h)^{-c}$ is equal to a, where c is computed as $\mathcal{H}(h, a, I)$.

The certificate is in effect an Okamoto signature on h made with a secret key that corresponds to the public key h_0h.

Key generation means of $\mathcal{U}$: The discussion of key pairs for $\mathcal{U}$ provided earlier with respect to the first secret-key certificate for the second preferred embodiment, applies here as well. (As will be clear to those of ordinary skill in the art, the symbols x_1, x_2 chosen here for convenience, do not refer to the symbols in that discussion.)

Certificate issuing means: Those of ordinary skill in the art are believed to be capable of straightforwardly applying (a) the inventive techniques for the issuing protocols of FIGS. **8** to **12**, and (b) the inventive technique to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$, to construct similar certificate issuing protocols for the present secret-key certificate.

3. Third Exemplary Secret-Key Certificate.

A third exemplary secret-key certificate in the second preferred embodiment, constructed by applying the general construction technique to the Fiat/Shamir signature scheme (See, Fiat, A. and Shamir, A., "How to prove yourself: practical solutions to identification and signature problems," Proceedings of Crypto '86, Lecture Notes in Computer Science 263, Springer-Verlag (1987), pp. 186--194), will now be described.

Key generation means of the KAC: The secret key of the CA is a tuple $(x_1,\ldots,x_k)$, where each x_i is in $\mathbb{Z}_n^*$. The corresponding public key is $(h_1,\ldots,h_k)$, where each h_i is equal to $(x_i^{-1})^2$. (Instead of squares, other powers can be used as well.)

The tuple $(h_1,\ldots,h_k,n)$ is made publicly known by the CA. The CA also makes publicly known a hash-function $\mathcal{H}$, which maps its arguments to, say, $\mathbb{Z}_{2^{kl}}$, for some appropriate security parameter l. This function should meet the requirements that are believed to make the Fiat/Shamir signature scheme secure.

Certificate verification means: A secret-key certificate on a public key $(h'_1,\ldots,h'_k)$ of $\mathcal{U}$, where each h'_i is in $\mathbb{Z}_n^*$, is a tuple $(c,r_1,\ldots,r_l)$, where c is in $\mathbb{Z}_{2^{kl}}$ and each r_i is in $\mathbb{Z}_n^*$, such that c is equal to $\mathcal{H}(\,(h'_1,\ldots,h'_k),(r_1^2\prod_{c_{1j}=1}h_jh'_j,\ldots,r_l^2\prod_{c_{lj}=1}h_jh'_j)\,)$. Here, the index j runs from 1 to k, and $(c_{i1},\ldots,c_{ik})$ is the binary vector consisting of the i-th group of k bits of the number c. The notation $\prod_{c_{ij}=1}h_jh_j$ denotes the product taken over all numbers $(h_jh'_j)$ with j such that the bit c_{ij} (this is the j-th bit in the i-th group of k bits of the number c) is equal to one.

One can consider the public key $(h_1,\ldots,h_k)$ of the CA to be a vector h_0, and the public key $(h'_1,\ldots,h'_k)$ of $\mathcal{U}$ to be a vector h. The secret-key certificate is in effect a Fiat/Shamir signature on h made with a secret key that corresponds, under the Fiat/Shamir signature scheme, to the public key h_0h, where the vector multiplication h_0h is defined by pairwise multiplication: h_0h is equal to $(h_1h'_1,\ldots,h_kh'_k)$.

Key generation means of $\mathcal{U}$: The key pair of $\mathcal{U}$ is of the same type as that of the CA. More precisely, the secret key corresponding to the public key $(h'_1,\ldots,h'_k)$, denoted by h', is a vector $(x'_1,\ldots,x'_k)$ such that $(x'_i)^{-2}=h'_i$.

Certificate issuing means: Those of ordinary skill in the art are believed to be capable of straightforwardly applying the inventive techniques for (a) the issuing protocols of FIGS. **8** to **12**, and (b) the inventive technique to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$, to construct similar certificate issuing protocols for the present secret-key certificate.

4. Fourth Exemplary Secret-Key Certificate.

Yet another Fiat/Shamir type signature scheme is the Feige/Fiat/Shamir signature scheme (See, Feige, U., Fiat, A. and Shamir, A., "Zero-knowledge proofs of identity," Journal of Cryptology 1 (1988), pp. 77--94). This scheme is a modification of the Fiat/Shamir scheme. Since the application of the general construction technique to this scheme is highly similar to the construction of the third exemplary secret-key certificate in the second preferred embodiment, a detailed description is omitted here. Again, for (a) each of the issuing protocols of FIGS. **8** to **12**, and (b) the inventive technique to issue the secret-key certificate to $\mathcal{U}$ and an additional party $\mathcal{T}$, a similar issuing protocol for the present secret-key certificate can be constructed straightforwardly in this manner.

5. Fifth Exemplary Secret-Key Certificate.

As in the first preferred embodiment, it will now be demonstrated that certain variations of the general construction technique can be used as well. Yet another exemplary secret-key certificate in the first preferred embodiment, constructed by applying a variation of the general construction technique to the Schnorr signature scheme, will now be described in detail.

Key generation means of the KAC: This is the same as in the description of the first secret-key certificate.

Certificate verification means: A secret-key certificate on a public key h in $\mathbb{Z}_n^*$ of $\mathcal{U}$ will now be taken to be a pair (c, r) in $\mathbb{Z}_{2^t} \times \mathbb{Z}_n^*$ such that c is equal to $\mathcal{H}(h, r^v h^{-c}, I)$.

The secret-key certificate can alternatively be taken to be a pair (a, r) in $\mathbb{Z}_n^* \times \mathbb{Z}_n^*$ such that $r^v h^{-c}$ is equal to a, where c is computed as $\mathcal{H}(h, a, I)$.

Key generation means of $\mathcal{U}$: In general, for the modified secret-key certificate to be secure, the public key of $\mathcal{U}$ must be defined as a product $g_1^{x_1} \cdots g_k^{x_k} h_0^{x_{k+1}} z_1^v$. (Those of ordinary skill in the art may wish to consider an even more general form, $(x_1, \ldots, x_k; x_{k+1}, z_1, \ldots, z_l)$, such that h is equal to $h_0^{x_{k+1}} \prod_{i=1}^k g_i^{x_i} \prod_{i=1}^l z_i^{v_i}$ for appropriate exponents v_i.) As with the first secret-key certificate, $g_1, \ldots, g_k$ are randomly chosen elements of large order in $\mathbb{Z}_n^*$, that are published by the CA in addition to v, h_0, n, and the description of $\mathcal{H}$. At most the CA should know the v-th root of each of $g_1, \ldots, g_k$.

In practice, one may want to use a simpler form of key pair. The simplest form is one in which the secret key of $\mathcal{U}$ is a number x in $\mathbb{Z}_v$, and the public key h is equal to $(h_0)^x$, and h may not be equal to 1. Another simple form is one in which the secret key of $\mathcal{U}$ is a pair $(x_1; x_2)$, such that h is equal to $h_0^{x_1} x_2^v$.

As in the first preferred embodiment, all the issuing techniques provided for the first secret-key certificate in the first preferred embodiment can be applied straightforward to construct issuing protocols for the modified secret-key certificate. It is believed that those of ordinary skill in the art are easily capable of doing so by studying the inventive techniques in conjunction with the flowcharts.

Instead, the following two remarks are made here, to help those of ordinary skill in the art appreciate the advantages of applying the general construction technique over applying the present variant thereof.

(1) Consider the restrictive blind secret-key certificate issuing protocol for the present secret-key certificate, which is similar to the flowchart of FIG. 7; if the secret key of $\mathcal{U}$ is a pair (I, x) in $\mathbb{Z}_v \times \mathbb{Z}_v$, and the public key h is equal to $h_0^I g_1^x$, then the number $I/x \bmod v$ can be encoded by the CA into the secret key ($\mathcal{U}$ will not be able to modify this quotient). However, for key pairs for $\mathcal{U}$ of this form, no secure signature schemes and proofs of knowledge are known in the art. On the

other hand, using a key pair for $\mathcal{U}$ such that the secret key is a pair $(I; x)$ in $\mathbb{Z}_v \times \mathbb{Z}_n^*$, and the public key is $h_0^I x^v$, has the problem that the information that has been encoded into the secret key by the CA, cannot be efficiently reconstructed from I and x. Hence, in order for the issuing protocol to have practical value, one must take the secret key of $\mathcal{U}$ to be a triple $(I_1, I_2; x)$ such that h is equal to $h_0^{I_1} g_1^{I_2} x^v$. This is less efficient than the key pair used in the restrictive blind issuing protocol for the first secret-key certificate.

(2) As will be appreciated, the security of the systems constructed using the general construction technique is closer related to the security of the underlying Fiat/Shamir signature scheme.

Conclusion.

This concludes the detailed descriptions of two preferred embodiments. While these descriptions of the present invention have been given as examples, it will be appreciated that various modifications, alternate configurations, and equivalents may be employed without departing from the spirit and scope of the present invention. For example, there are many essentially equivalent orders to evaluate expressions; ways to evaluate expressions; ways to order expressions, tests, and transmissions within flowchart boxes; ways to group operations into flowchart boxes; and ways to order flowchart boxes. The particular choices that have been made here are merely for clarity in exposition.

Certain variations and substitutions may be apparent to those of ordinary skill in the art. Although various such variations and substitutions have been indicated and sometimes described in detail in the text, this may be more fully appreciated in the light of the following examples.

First, the exemplary secret-key certificates that have been described are derived from Fiat/Shamir type signature schemes by the general following construction technique: denoting the public key of $\mathcal{U}$ by h, and that of the CA by h_0, a secret-certificate on h in effect is a signature of the underlying Fiat/Shamir type on the message h made with a

secret key that corresponds to public key h_0h. It has already been shown, at the end of the description of the first preferred embodiment, that variations on the general construction technique may be applied as well. As will be appreciated, the particular form h_0h in the general construction technique is not essential. Taking, for example, the form h_0h^k for a fixed integer k different from 1 would obviously work as well. The essence of the general construction technique is that the secret-key certificate is constructed from any Fiat/Shamir type signature scheme by letting the certificate in effect be a signature on the public key of $\mathcal{U}$, where the signature is of the underlying Fiat/Shamir type signature scheme and made with a secret key that corresponds, under the Fiat/Shamir type signature scheme, to a public key which is a suitable mathematical function of the public key of $\mathcal{U}$ and the public key of the CA.

Second, hierarchic certification can be implemented with secret-key certificates. As will be clear to those of ordinary skill in the art, $\mathcal{U}$ in turn can use his certified key pair to issue a secret-key certificate to another party, and so on. In this way, a hierachical certification tree can be constructed: each node in this tree can be considered to be a pair consisting of a public key and a secret-key certificate on the public key, where a parent node certifies the key pairs of its child nodes by issuing a secret-key certificate on the public key of each child node, computed by using a secret key corresponding to the public key of the parent node. If, for instance, a decryption can be performed by a party associated with a node in the tree, then this party must know the secret key corresponding to the public key in that node; this in turn implies that the secret-key certificate must have been computed by the party associated with the parent node, and so this party in turn knows the secret key corresponding to the public key of the parent node; hence, the secret-key certificate of the parent node must have been computed by the party associated with the parent node of the parent node; and so on, all the way to the root node.

Third, the secret-key certificate technique can be used to

construct secure digital signature schemes. To sign a message, the signer party, which has a first secret key and a matching first public key, first generates independently at random a one-time secret key and a matching one-time public key, where the word "one-time" is used only to emphasize that these keys will be used only once by the signer party. The signer party then computes a secret-key certificate on the one-time public key with respect to the first public key. It can do this because it knows the first public key. It then signs the message with respect to the one-time public key, which it can do because it knows the one-time secret key. The resulting signature of the signer party on the message consists of the one-time public key, the secret-key certificate on the one-time public key, and the signature on the message with respect to the one-time public key. To verify the signature, the certificate verification means are used to verify the secret-key certificate, and the signature on the message is verified by using the one-time public key. The signer party can generate independently at random a new one-time secret key and matching one-time public key each time that it has to sign a message. In effect, this method of constructing a secure digital signature scheme using the secret-key certificate issuing technique is the same as that applied in the example of hierarchic certification, disclosed in the preceding paragraph.

It will also be obvious to those of ordinary skill in the art how parts of the inventive techniques and protocols disclosed here can be used to advantage.

WHAT IS CLAIMED IS:

1. A cryptographic method wherein a first party issues a certificate, called a secret-key certificate, to a second party, the method comprising the steps of:

   generating, for use by the first party, a first secret key, unknown at least in part to the second party, and a corresponding first public key;

   generating, for use by the second party, a second secret key and a corresponding second public key;

   issuing by the first party to the second party, in a certificate issuing protocol, a secret-key certificate certificate corresponding to the second public key according to a publicly verifiable relation, the secret-key certificate being a digital signature of the first party on the second secret key, and the second party being able to feasibly generate without assistance of the first party public keys and corresponding secret-key certificates.

2. A method as in claim **1**, where the second public key, the corresponding secret-key certificate and information related to the second party are listed in a public-key directory.

3. A method as in claim **1**, where the second party uses the second secret key to perform at least one of the following cryptographic actions, namely, digital signing; proving knowledge of a secret key corresponding to the second public key according to the public-key scheme, without revealing the second secret key; decrypting a message that is encrypted with the second public key; or issuing a secret-key certificate corresponding to a public key.

4. A method as in claim **1**, where the first party signs a message for a third party by:

   generating at random a one-time secret key and a corresponding one-time public key;

computing a secret-key certificate corresponding to the one-time public key, by applying the first secret key; and

computing a digital signature on the message, with respect to the one-time public key, by applying the one-time secret key,

the digital signature of the first party on the message consisting of the one-time public key, the secret-key certificate corresponding to the one-time public key and the digital signature on the message that has been made with respect to the one-time public key.

5. A method as in claim **1**, where the second party in the certificate issuing protocol does not reveal the second secret key to the first party.

6. A method as in claim **1**, where the second party in the certificate issuing protocol blinds the second secret key, the second public key and the corresponding secret-key certificate.

7. A method as in claim **1**, where the second party in the certificate issuing protocol blinds the second public key and the corresponding secret-key certificate, but not a pre-determined non-constant function of the second secret key.

8. A method as in claim **1**, where the second party comprises a first sub-party, that acts in the interest of the first party, and a second sub-party, the second secret key not being known to either one of the two sub-parties.

9. A method as in claim **1**, where the secret-key certificate is a digital signature key of the Fiat/Shamir type, made with respect to a public key that is a combination of the first public key and the second public key.

10. A method as in claim **9**, where the digital signature is a Schnorr digital signature.

11. A method as in claim **1**, where the second secret key represents a set of credentials of the second party.

12. A method as in claim **1**, where the secret-key certificate represents a credential issued by the first party to the second party.

13. Apparatus for implementing a cryptographic system in which a first party issue a certificate, called a secret-key certificate, to a second party, the apparatus comprising:

first key generation means that, on being given as input at least a security parameter, outputs a pair consisting of a first secret key and a corresponding first public key, for use by the first party;

second key generation means that, on being given as input at least a security parameter, outputs a pair consisting of a second secret key and a corresponding second public key, for use by the second party;

certificate verification means that, on being given as input the first public key and a pair consisting of a third public key and a presumed secret-key certificate corresponding to the third public key, responds affirmatively or negatively, depending on whether the presumed certificate corresponds to the third public key or not;

certificate issuing means that, on being given as input the first secret key, the second secret key and the second public key, outputs a secret-key certificate corresponding to the second public key, the secret-key certificate being a digital signature of the first party on the second secret key; and

certificate simulating means that, on being given as input the first public key, outputs a fourth public key and a secret-key certificate corresponding to the fourth public key, the probability distribution of the output of the certificate simulating means being substantially indistinguishable from the probability

distribution that applies when a public key is generated by the second key generation means and a corresponding secret-key certificate is generated by the certificate issuing means.

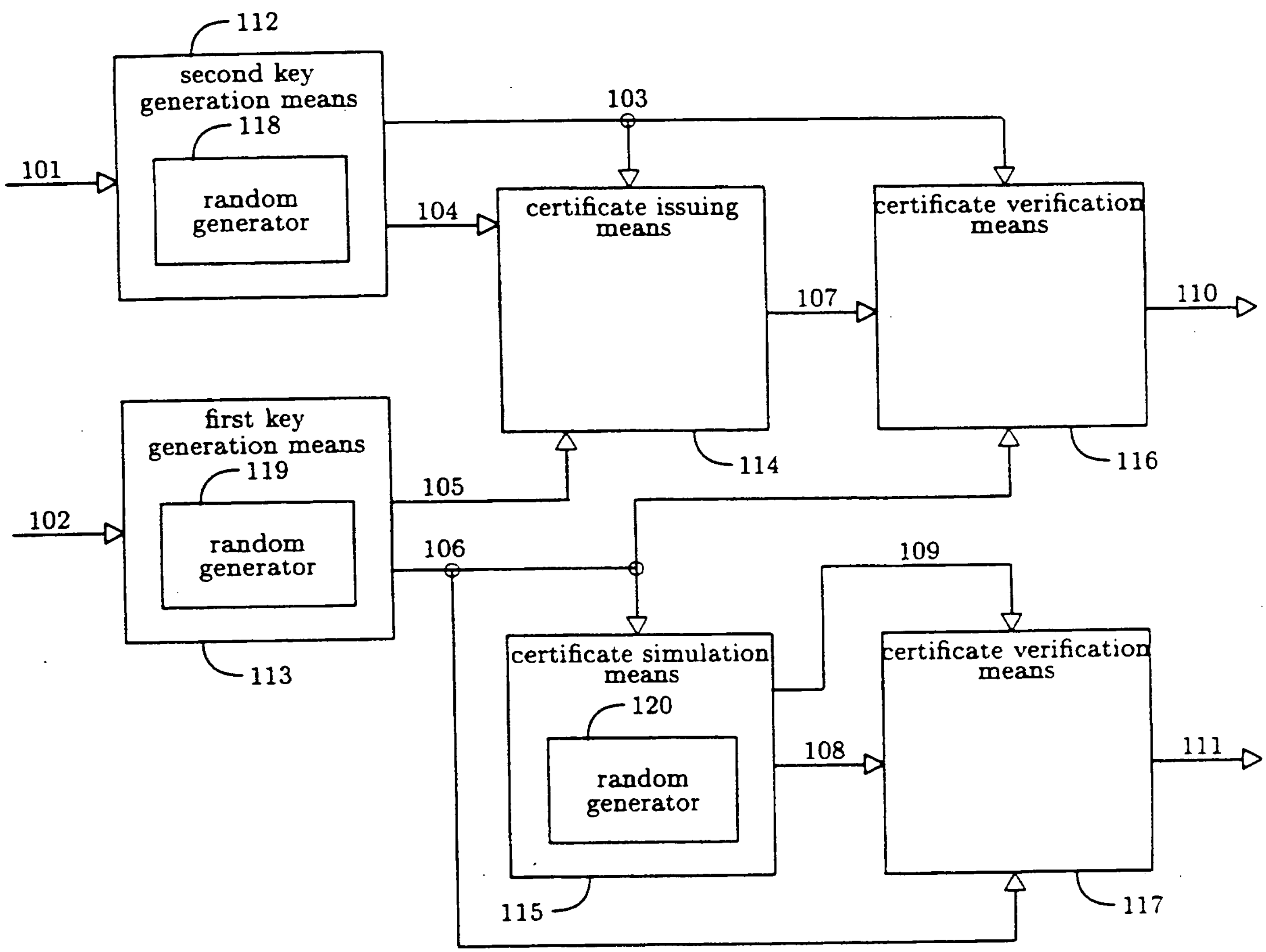

112
second key generation means
118
random generator
101
104
103
certificate issuing means
certificate verification means
107
110
114
116
first key generation means
119
random generator
102
105
106
113
109
certificate simulation means
120
random generator
certificate verification means
108
111
115
117

FIGURE 1

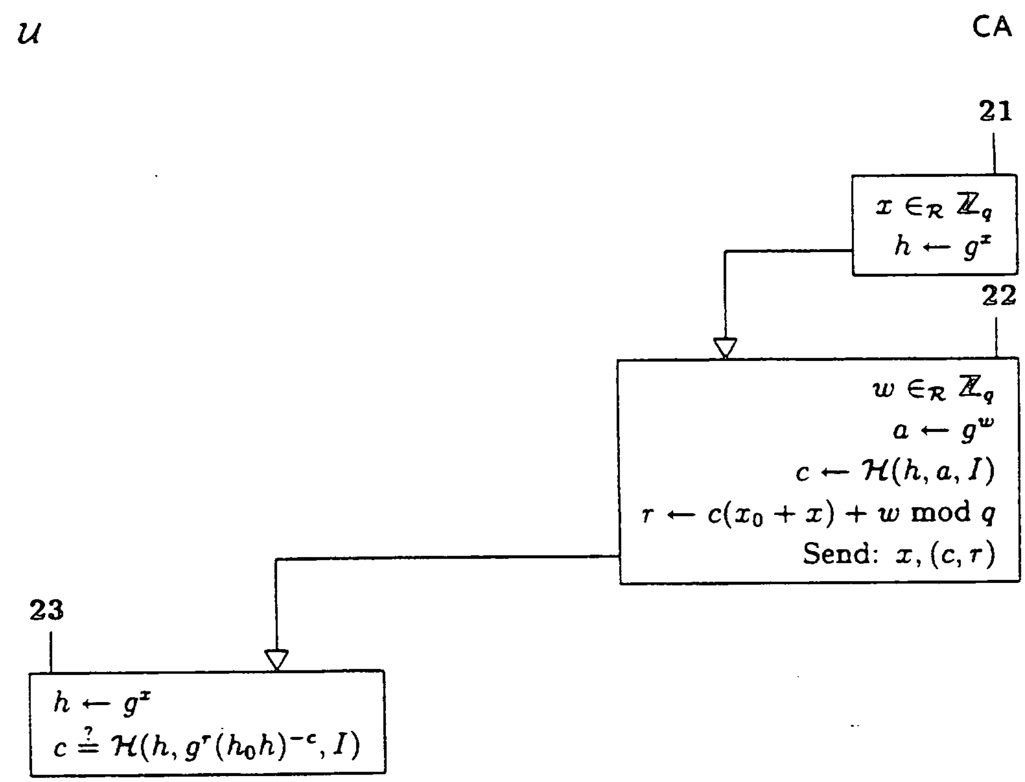

$\mathcal{U}$
CA
21
$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow g^x$
22
$w \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow g^w$
$c \leftarrow \mathcal{H}(h, a, I)$
$r \leftarrow c(x_0 + x) + w \bmod q$
Send: $x, (c, r)$
23
$h \leftarrow g^x$
$c \stackrel{?}{=} \mathcal{H}(h, g^r(h_0 h)^{-c}, I)$

FIGURE 2

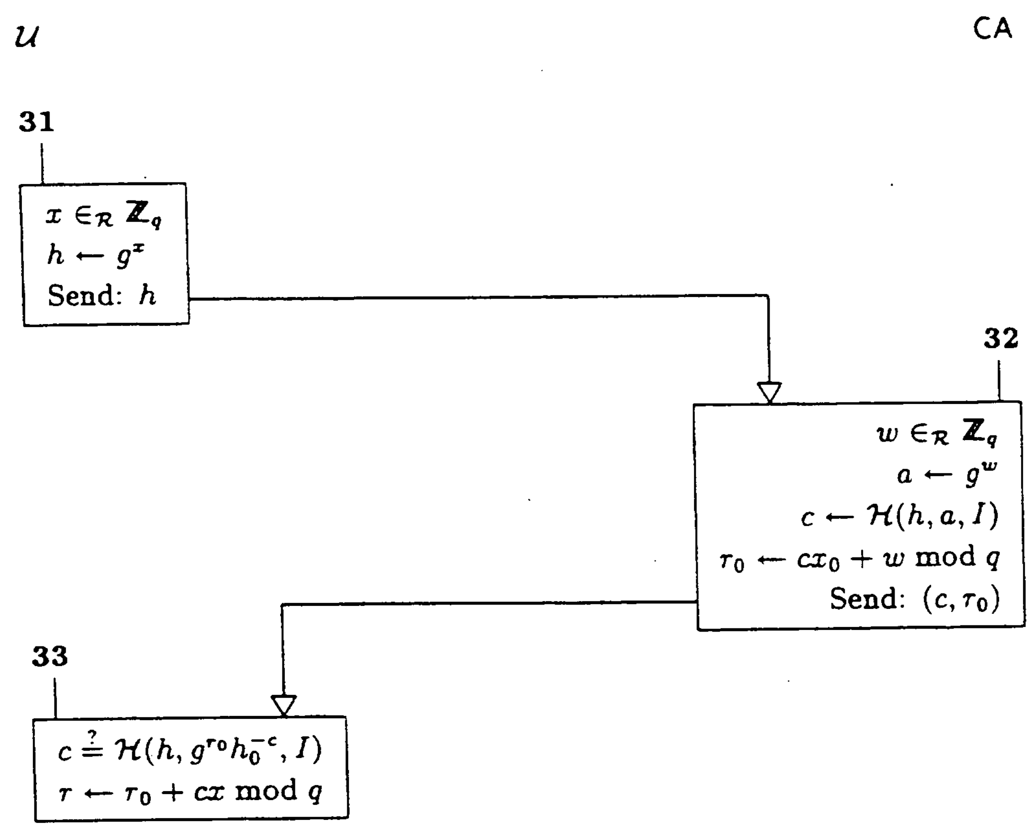

$\mathcal{U}$
CA
31
$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow g^x$
Send: h
32
$w \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow g^w$
$c \leftarrow \mathcal{H}(h, a, I)$
$r_0 \leftarrow cx_0 + w \bmod q$
Send: (c, r_0)
33
$c \stackrel{?}{=} \mathcal{H}(h, g^{r_0} h_0^{-c}, I)$
$r \leftarrow r_0 + cx \bmod q$

FIGURE 3

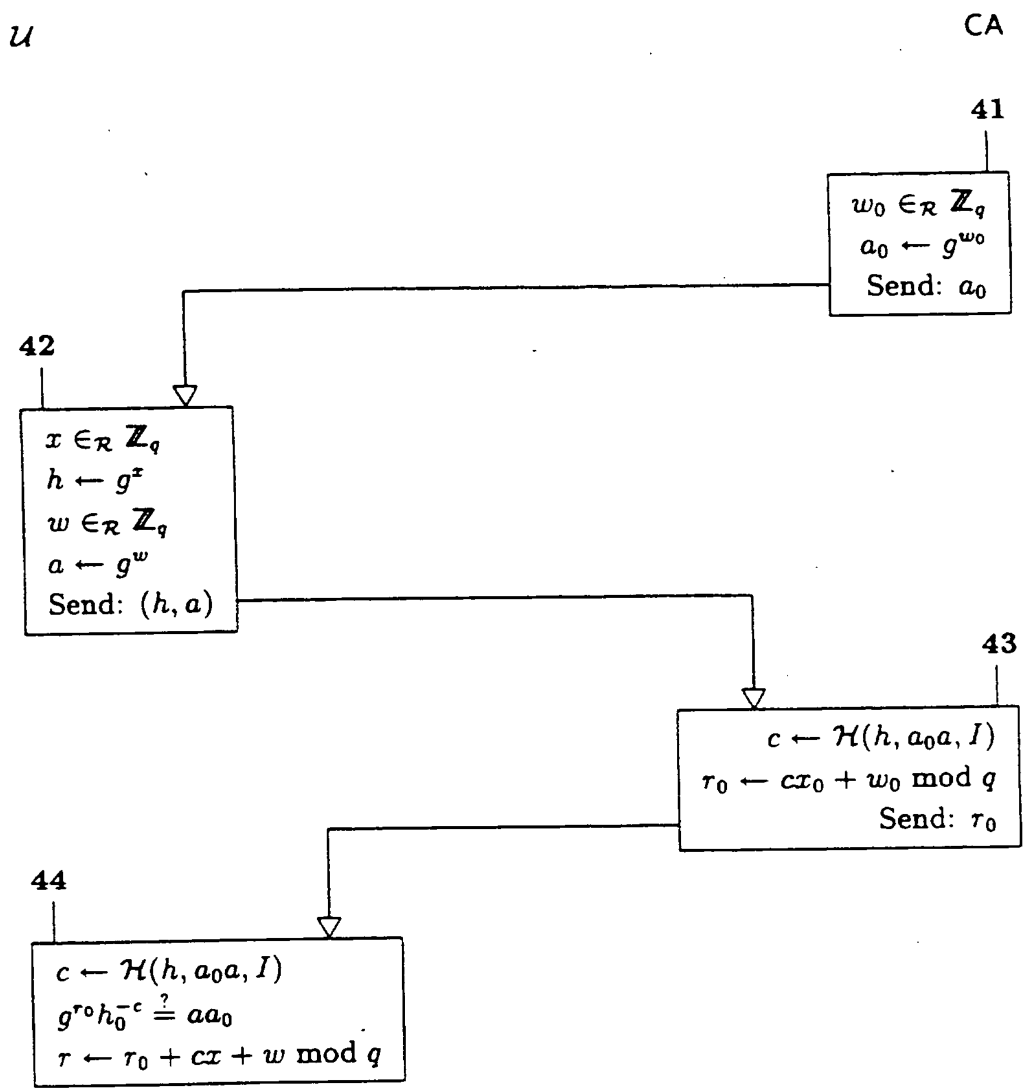

$\mathcal{U}$
CA
41
$w_0 \in_{\mathcal{R}} \mathbb{Z}_q$
$a_0 \leftarrow g^{w_0}$
Send: a_0
42
$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow g^x$
$w \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow g^w$
Send: (h, a)
43
$c \leftarrow \mathcal{H}(h, a_0 a, I)$
$r_0 \leftarrow c x_0 + w_0 \bmod q$
Send: r_0
44
$c \leftarrow \mathcal{H}(h, a_0 a, I)$
$g^{r_0} h_0^{-c} \stackrel{?}{=} a a_0$
$r \leftarrow r_0 + cx + w \bmod q$

FIGURE 4

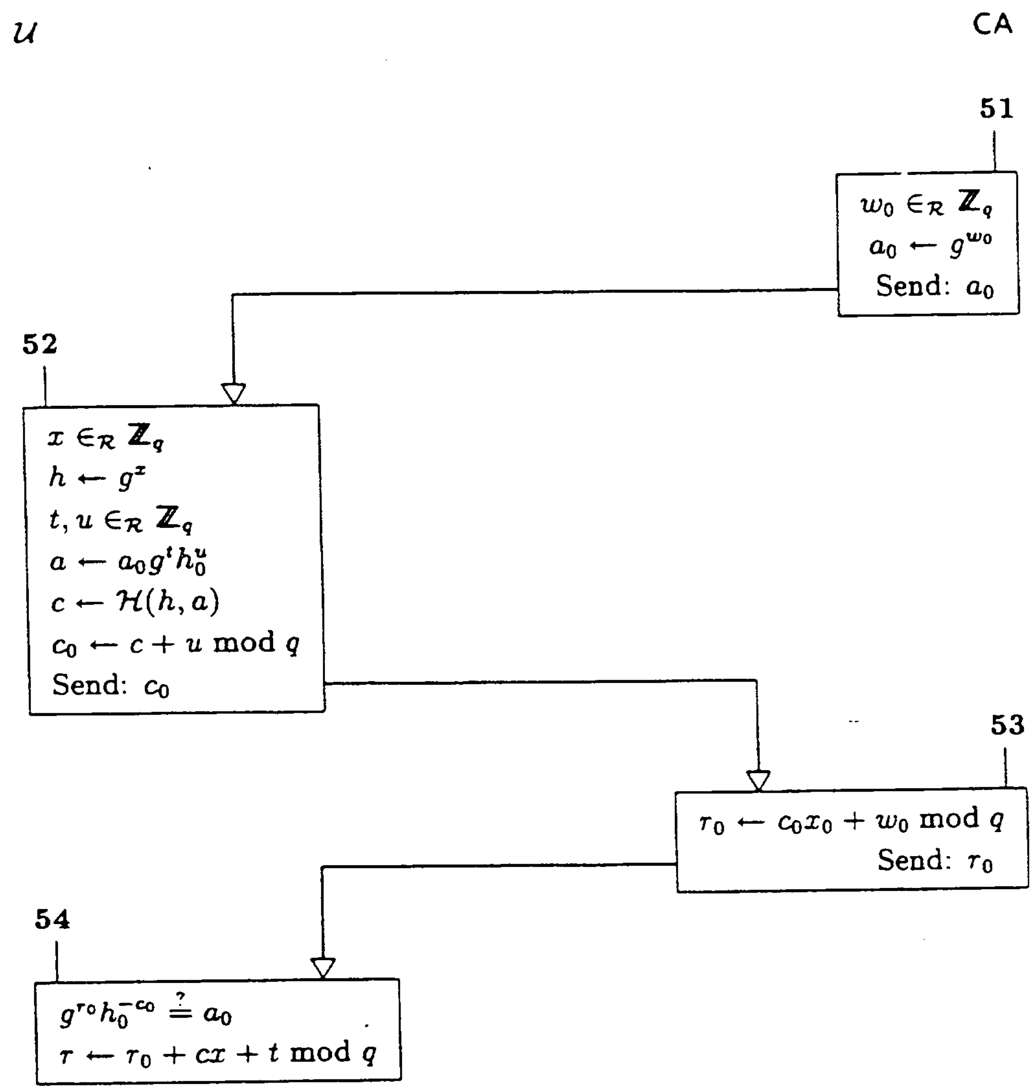

$\mathcal{U}$
CA
51
$w_0 \in_{\mathcal{R}} \mathbb{Z}_q$
$a_0 \leftarrow g^{w_0}$
Send: a_0
52
$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow g^x$
$t, u \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow a_0 g^t h_0^u$
$c \leftarrow \mathcal{H}(h, a)$
$c_0 \leftarrow c + u \bmod q$
Send: c_0
53
$r_0 \leftarrow c_0 x_0 + w_0 \bmod q$
Send: r_0
54
$g^{r_0} h_0^{-c_0} \stackrel{?}{=} a_0$
$r \leftarrow r_0 + cx + t \bmod q$

FIGURE 5

$\mathcal{U}$ CA

61

$w_0 \in_{\mathcal{R}} \mathbb{Z}_q$
$a_0 \leftarrow g^{w_0}$
Send: a_0

62

$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow g^x g_1^I$
$t, u \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow a_0 g^t (h_0 g_1^I)^u$
$c \leftarrow \mathcal{H}(h, a)$
$c_0 \leftarrow c + u \bmod q$
Send: c_0

63

$r_0 \leftarrow c_0(x_0 + yI) + w_0 \bmod q$
Send: r_0

64

$g^{r_0}(h_0 g_1^I)^{-c_0} \stackrel{?}{=} a_0$
$r \leftarrow r_0 + cx + t \bmod q$

FIGURE 6

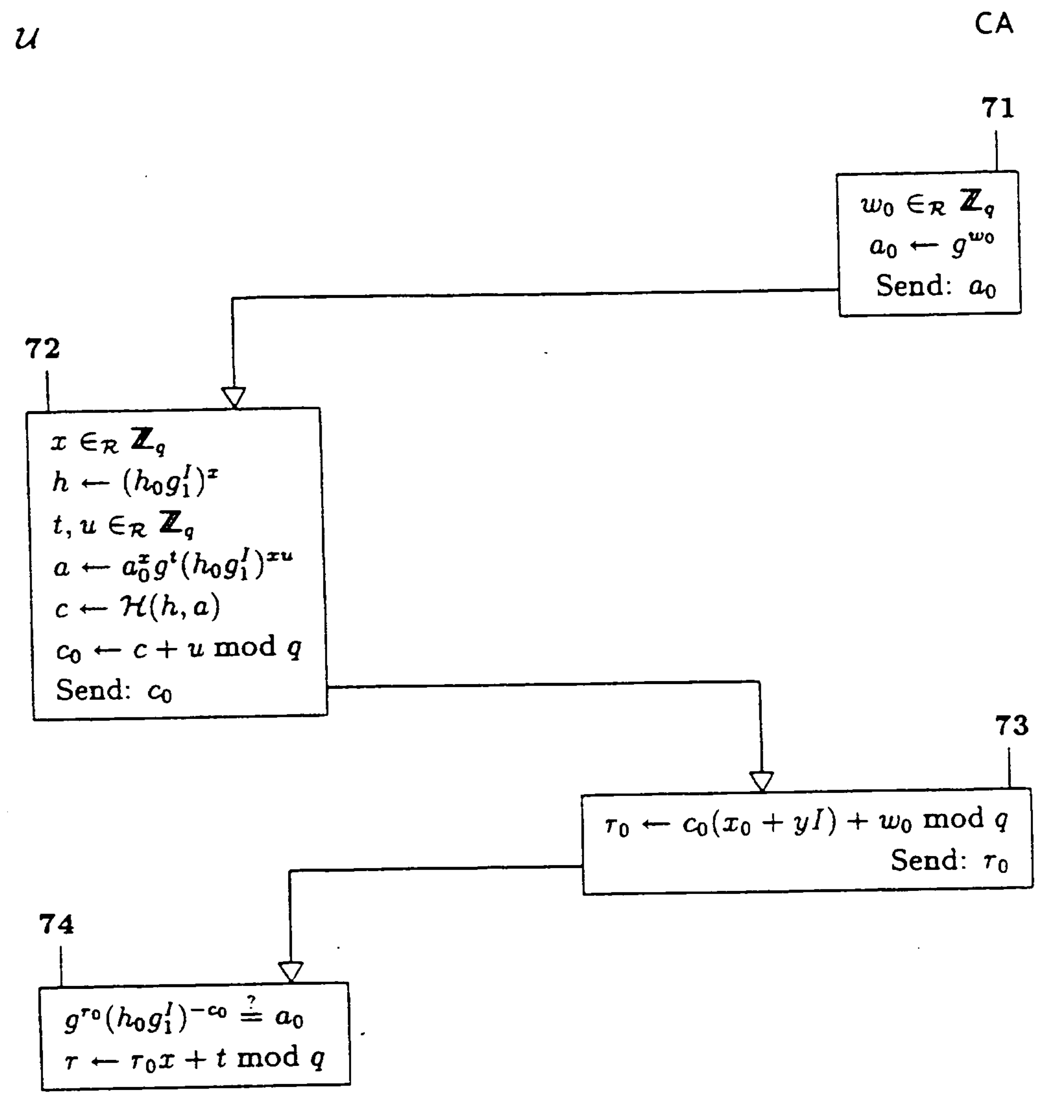

$\mathcal{U}$
CA
71
$w_0 \in_{\mathcal{R}} \mathbb{Z}_q$
$a_0 \leftarrow g^{w_0}$
Send: a_0
72
$x \in_{\mathcal{R}} \mathbb{Z}_q$
$h \leftarrow (h_0 g_1^I)^x$
$t, u \in_{\mathcal{R}} \mathbb{Z}_q$
$a \leftarrow a_0^x g^t (h_0 g_1^I)^{xu}$
$c \leftarrow \mathcal{H}(h, a)$
$c_0 \leftarrow c + u \bmod q$
Send: c_0
73
$r_0 \leftarrow c_0(x_0 + yI) + w_0 \bmod q$
Send: r_0
74
$g^{r_0}(h_0 g_1^I)^{-c_0} \stackrel{?}{=} a_0$
$r \leftarrow r_0 x + t \bmod q$

FIGURE 7

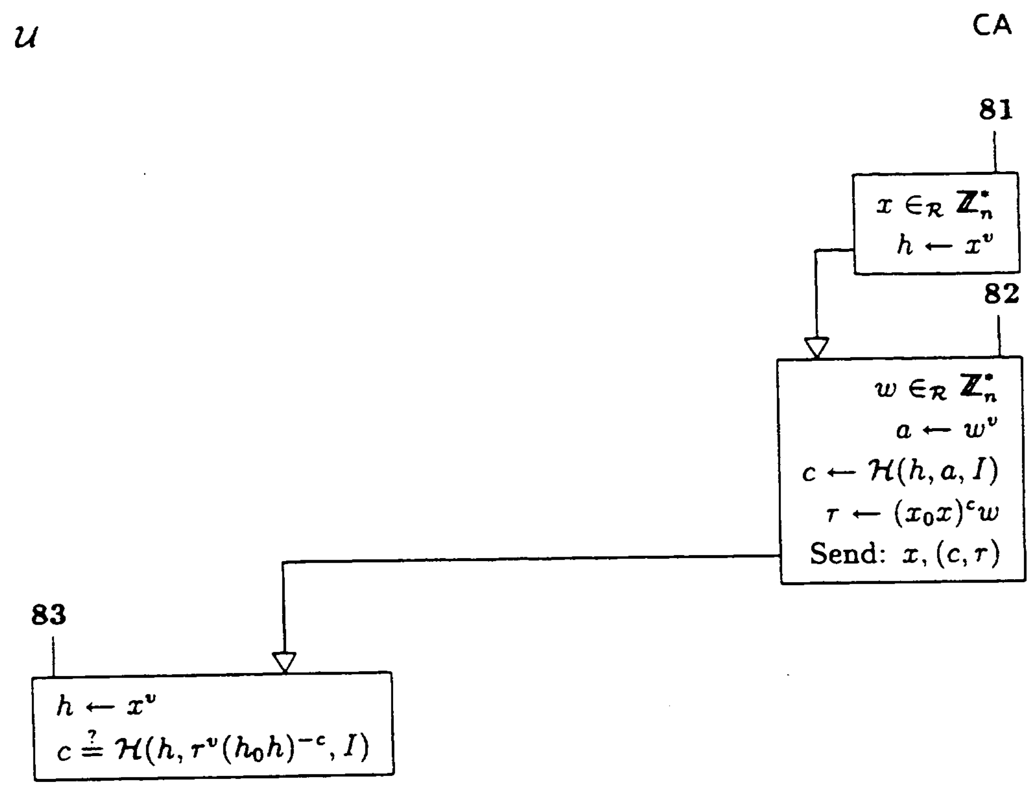

$\mathcal{U}$
CA
81
$x \in_{\mathcal{R}} \mathbb{Z}_n^*$
$h \leftarrow x^v$
82
$w \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a \leftarrow w^v$
$c \leftarrow \mathcal{H}(h, a, I)$
$r \leftarrow (x_0 x)^c w$
Send: $x, (c, r)$
83
$h \leftarrow x^v$
$c \stackrel{?}{=} \mathcal{H}(h, r^v (h_0 h)^{-c}, I)$

FIGURE 8

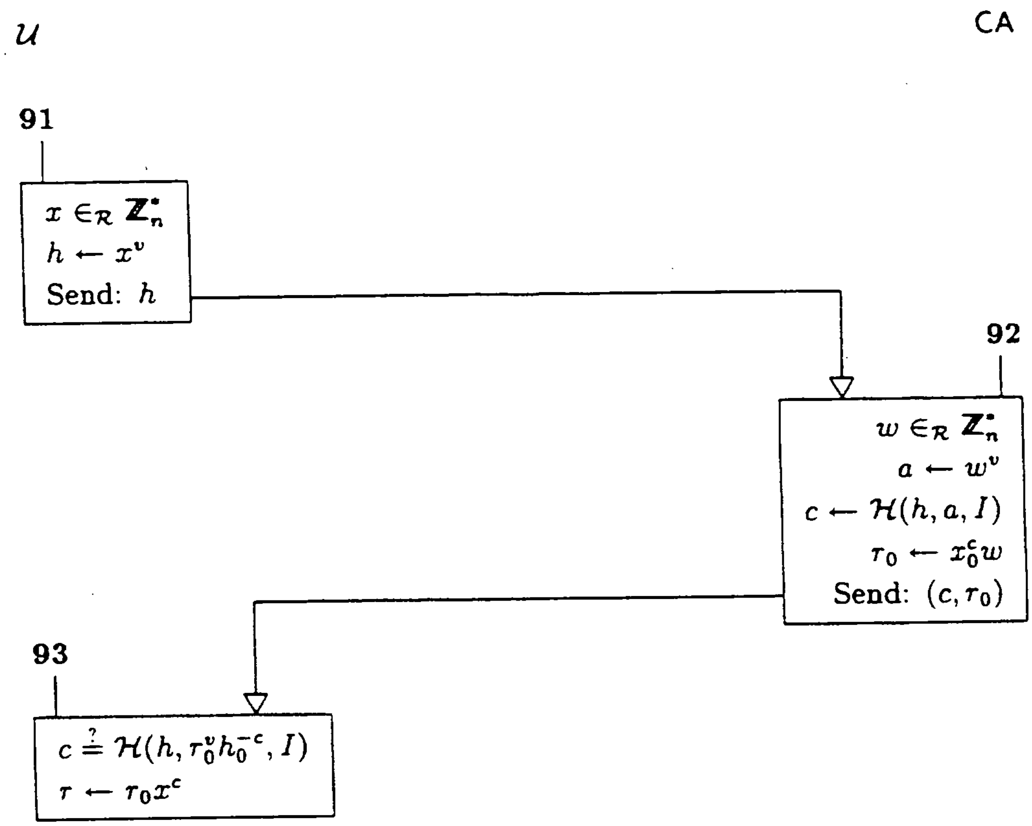

$\mathcal{U}$
CA
91
$x \in_{\mathcal{R}} \mathbb{Z}_n^*$
$h \leftarrow x^v$
Send: h
92
$w \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a \leftarrow w^v$
$c \leftarrow \mathcal{H}(h, a, I)$
$r_0 \leftarrow x_0^c w$
Send: (c, r_0)
93
$c \stackrel{?}{=} \mathcal{H}(h, r_0^v h_0^{-c}, I)$
$r \leftarrow r_0 x^c$

FIGURE 9

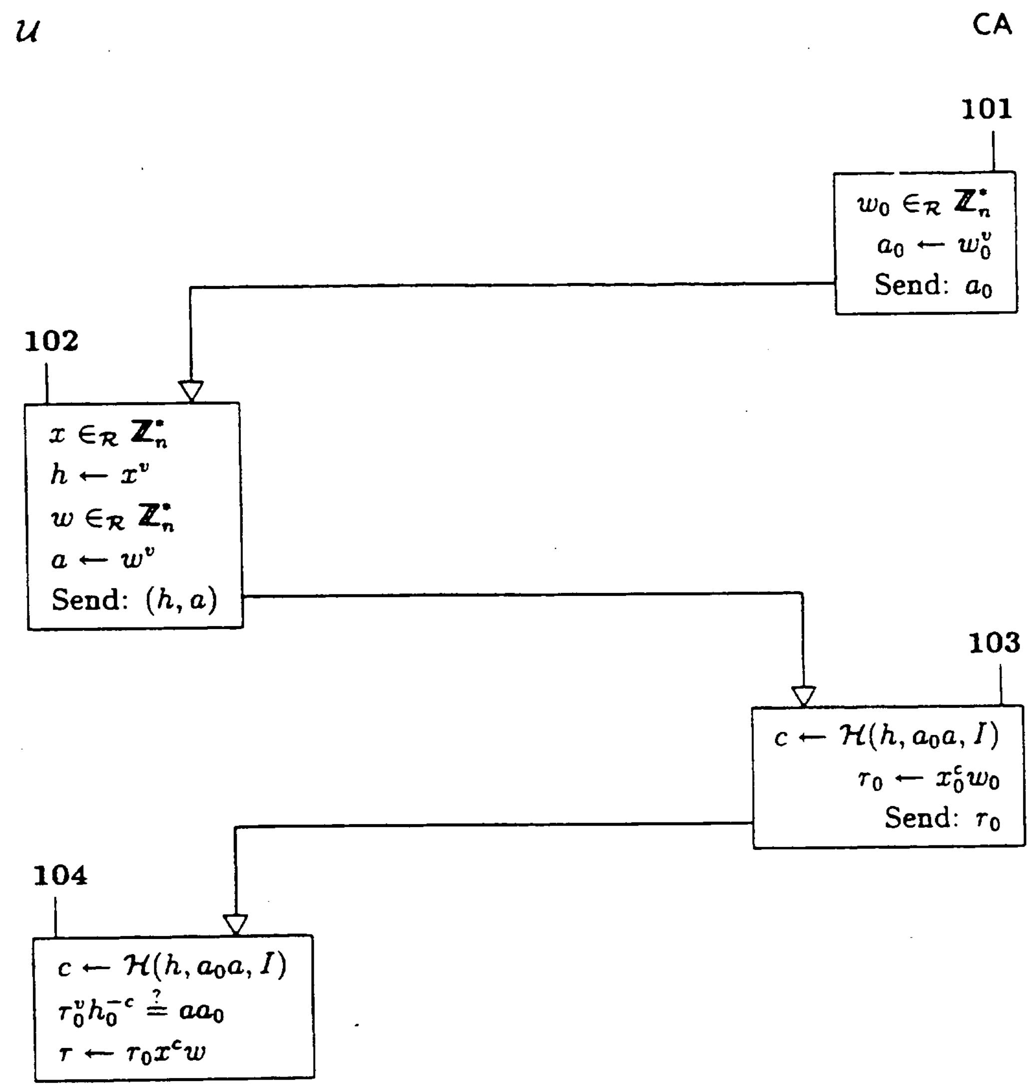

$\mathcal{U}$
CA
101
$w_0 \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a_0 \leftarrow w_0^v$
Send: a_0
102
$x \in_{\mathcal{R}} \mathbb{Z}_n^*$
$h \leftarrow x^v$
$w \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a \leftarrow w^v$
Send: (h, a)
103
$c \leftarrow \mathcal{H}(h, a_0 a, I)$
$r_0 \leftarrow x_0^c w_0$
Send: r_0
104
$c \leftarrow \mathcal{H}(h, a_0 a, I)$
$r_0^v h_0^{-c} \stackrel{?}{=} a a_0$
$r \leftarrow r_0 x^c w$

FIGURE 10

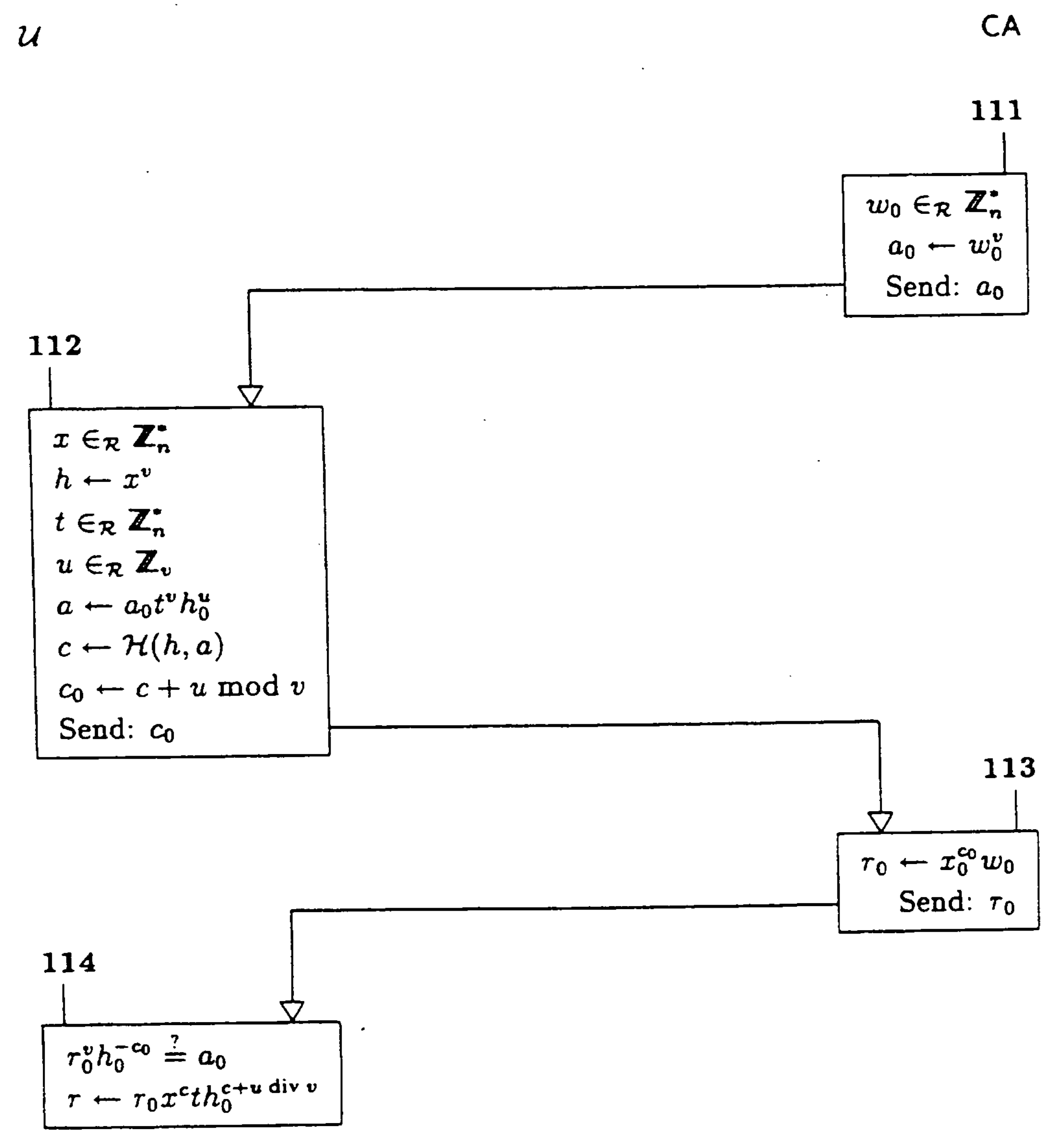

$\mathcal{U}$
CA
111
$w_0 \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a_0 \leftarrow w_0^v$
Send: a_0
112
$x \in_{\mathcal{R}} \mathbb{Z}_n^*$
$h \leftarrow x^v$
$t \in_{\mathcal{R}} \mathbb{Z}_n^*$
$u \in_{\mathcal{R}} \mathbb{Z}_v$
$a \leftarrow a_0 t^v h_0^u$
$c \leftarrow \mathcal{H}(h, a)$
$c_0 \leftarrow c + u \bmod v$
Send: c_0
113
$r_0 \leftarrow x_0^{c_0} w_0$
Send: r_0
114
$r_0^v h_0^{-c_0} \stackrel{?}{=} a_0$
$r \leftarrow r_0 x^c t h_0^{c+u \text{ div } v}$

FIGURE 11

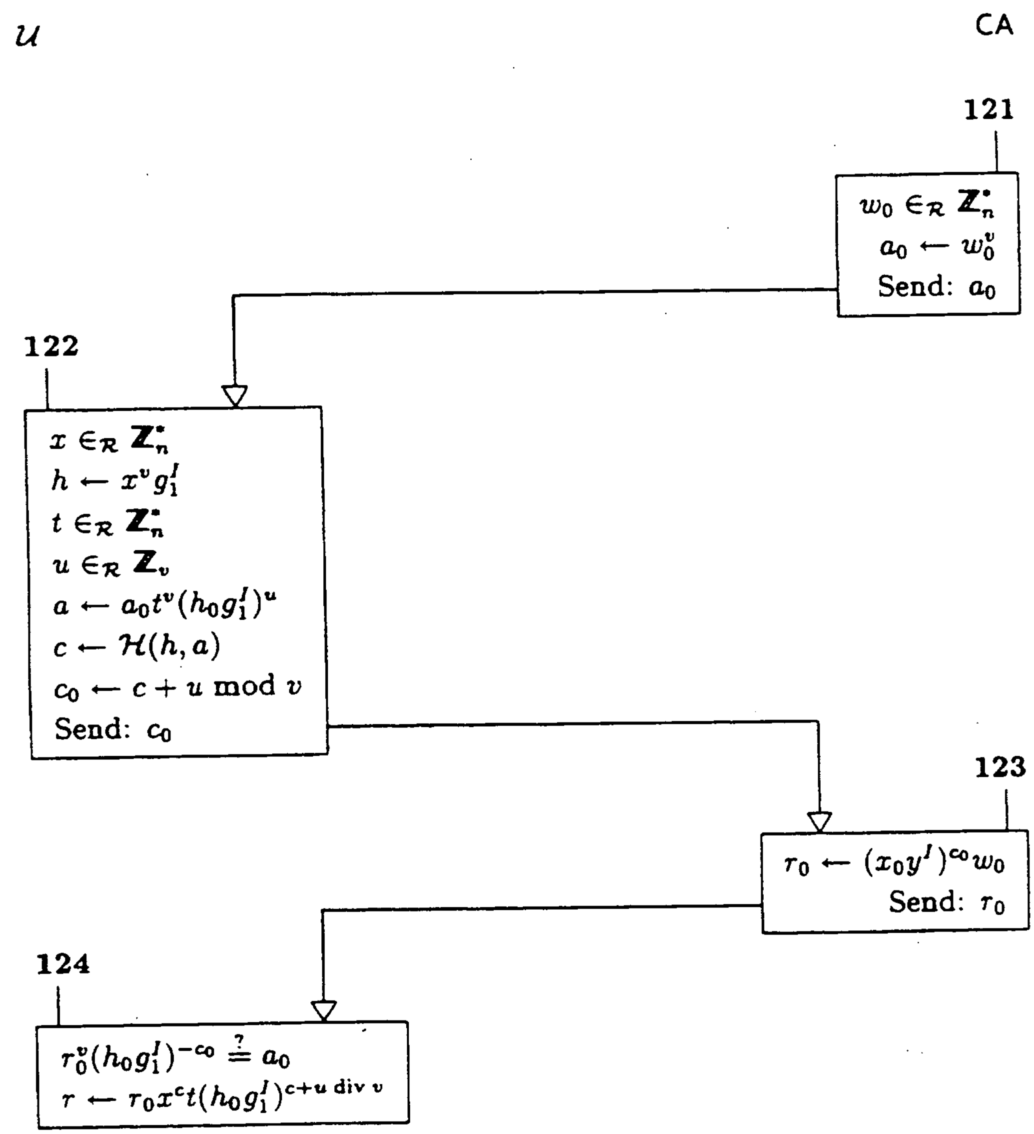

$\mathcal{U}$
CA
121
$w_0 \in_{\mathcal{R}} \mathbb{Z}_n^*$
$a_0 \leftarrow w_0^v$
Send: a_0
122
$x \in_{\mathcal{R}} \mathbb{Z}_n^*$
$h \leftarrow x^v g_1^I$
$t \in_{\mathcal{R}} \mathbb{Z}_n^*$
$u \in_{\mathcal{R}} \mathbb{Z}_v$
$a \leftarrow a_0 t^v (h_0 g_1^I)^u$
$c \leftarrow \mathcal{H}(h, a)$
$c_0 \leftarrow c + u \bmod v$
Send: c_0
123
$r_0 \leftarrow (x_0 y^I)^{c_0} w_0$
Send: r_0
124
$r_0^v (h_0 g_1^I)^{-c_0} \stackrel{?}{=} a_0$
$r \leftarrow r_0 x^c t (h_0 g_1^I)^{c+u \text{ div } v}$

FIGURE 12

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/32	A3	(11) International Publication Number: WO 96/12362 (43) International Publication Date: 25 April 1996 (25.04.96)

(21) International Application Number: PCT/NL95/00350

(22) International Filing Date: 12 October 1995 (12.10.95)

(30) Priority Data:
08/321,855 14 October 1994 (14.10.94) US

(71)(72) Applicant and Inventor: BRANDS, Stefanus, Alfonsus [NL/NL]; Ina Boudier-Bakkerlaan 143 III, NL-3582 XW Utrecht (NL).

(81) Designated States: AM, AU, BB, BG, BR, BY, CA, CN, CZ, EE, FI, GE, HU, IS, JP, KG, KP, KR, KZ, LK, LR, LT, LV, MD, MG, MN, MX, NO, NZ, PL, RO, RU, SG, SI, SK, TJ, TM, TT, UA, UG, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ, UG).

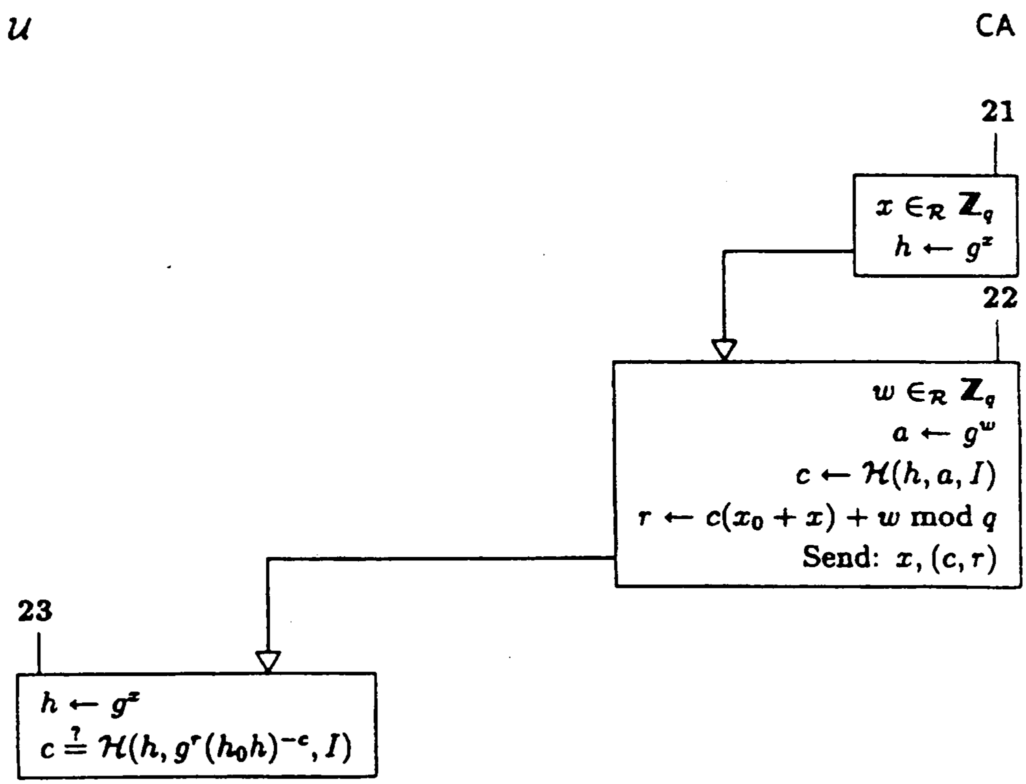
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(88) Date of publication of the international search report: 30 May 1996 (30.05.96)

(54) Title: SECRET-KEY CERTIFICATES

(57) Abstract

Cryptographic methods and apparatus are disclosed that enable formation and issuance of secret-key certificates. In contrast to the well-known cryptographic technique of public-key certificates, where a public-key certificate is a digital signature of a certification authority on a public key, pairs consisting of a public key and a corresponding secret-key certificate can be generated by anyone. However, triples consisting of a secret key, a matching public key and a secret-key certificate corresponding to the public key, can only be generated when the certification authority is involved. Also described are applications of secret-key certificates to public-key directories and to restrictive blind issuing protocols.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgystan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

INTERNATIONAL SEARCH REPORT

International Application No
PCT/NL 95/00350

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 139 313 (CHAUM) 2 May 1985 see abstract ---	1,2,13
A	ADVANCES IN CRYPTOLOGY (CRYPTO), 22 - 26 August 1993 BERLIN, DE, pages 302-318, BRANDS 'Untracable off-line cash in wallet with observers' see abstract ---	1,2,13
A	ADVANCES IN CRYPTOLOGY (CRYPTO), 21 - 25 August 1994 BERLIN, DE, pages 83-94, DEIOS & QUISQUATER 'An identity-based signature scheme with bounded life-span' see abstract --- -/--	1,2,13

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
28 March 1996	12.04.96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Scriven, P

Form PCT/ISA/210 (second sheet) (July 1992)

BNSDOCID: <WO 9612362A3>

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JOURNAL OF CRYPTOLOGY, vol. 4, no. 3, pages 161-174, SCHNORR 'Efficient signature generation by smart cards' see abstract ---	1,2,13
X,P	LATIN '95: THEORETICAL INFORMATICS. SECOND LATIN AMERICAN SYMPOSIUM, 3 - 7 April 1995 BERLIN, DE, pages 131-166, BRANDS 'Off-line electronic cash based on secret-key certificates' see abstract ---	1-13
X,P	EUROCRYPT '95. INTERNATIONAL CONFERENCE ON THE THEORY AND APPLICATION OF CRYPTOGRAPHIC TECHNIQUES, 21 - 25 May 1995 BERLIN, DE, pages 231-247, BRANDS 'Restrictive binding of secret-key certificates' see abstract -----	1-13

1

BNSDOCID: <WO 9612362A3>

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/NL 95/00350

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0139313	02-05-85	US-A- 4759063	19-07-88
		DE-A- 3485804	13-08-92
		US-A- 4926480	15-05-90

Form PCT/ISA/210 (patent family annex) (July 1992)

THIS PAGE BLANK (USPTO)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)